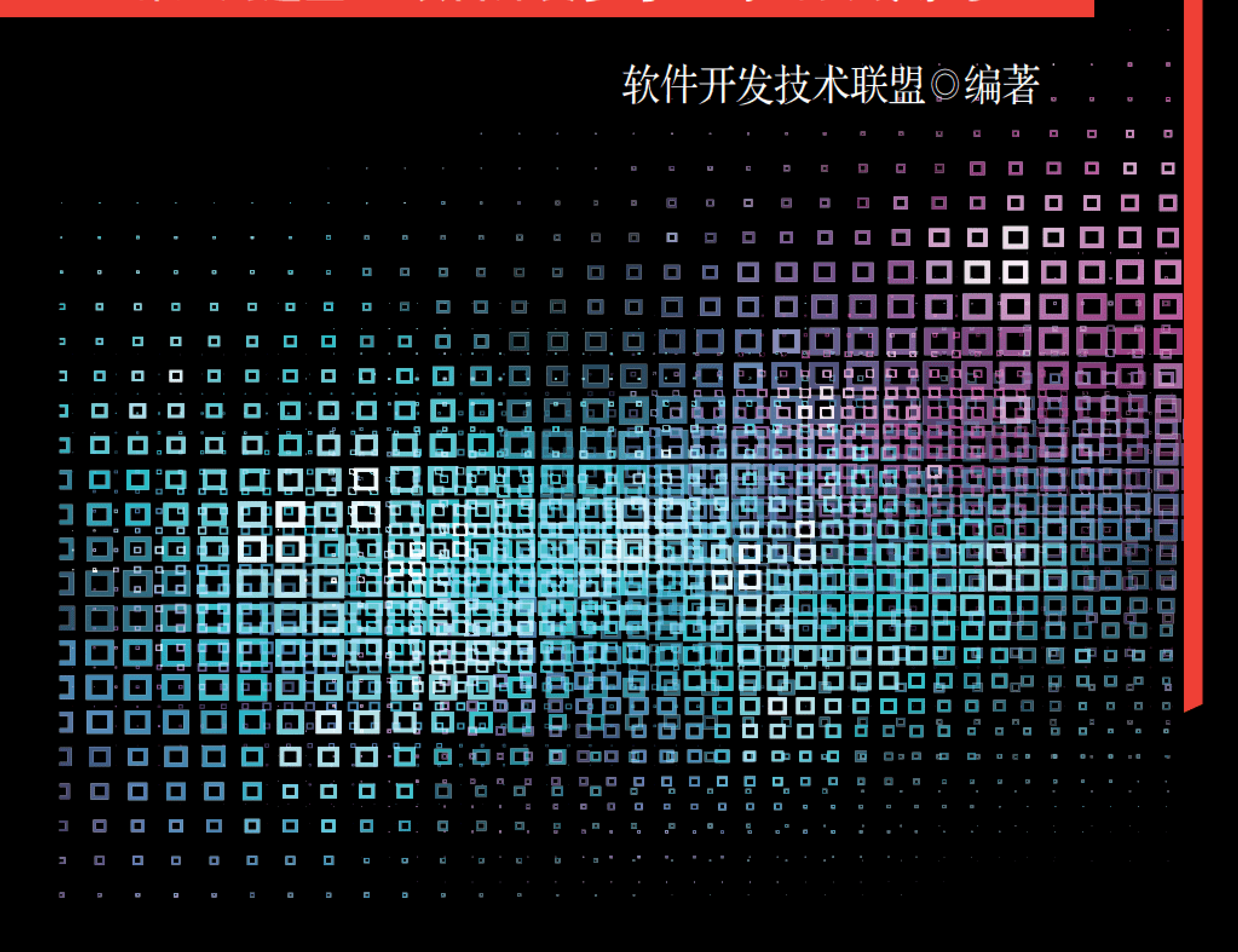
软件工程师开发大系

PHP 开发实例大全（基础卷）

625个典型实例及源码分析，涵盖15个应用方向

✔工作应用速查 ✔项目开发参考 ✔学习实战练习

软件开发技术联盟◎编著



- 工作应用速查：本书实例全面、系统，涉及程序开发的各个方面，适合各级程序开发人员速查速用。
- 项目开发参考：程序员借助本书提供的实例源代码，可以快速搭建工程项目，提高开发效率。
- 学习实战练习：入门者的实战训练大全，不但可以激发学习兴趣，更可提高编程实战能力和编程思维水平。



清华大学出版社

软件工程师开发大系

PHP 开发实例大全

（基础卷）

软件开发技术联盟　编著

清华大学出版社

北　京

内 容 简 介

《PHP 开发实例大全（基础卷）》筛选、汇集了 PHP 开发从基础知识到高级应用各个层面的大量实例及源代码，共有 600 个左右，每个实例及源代码按实例说明、关键技术、设计过程、详尽注释、秘笈心法的顺序进行了分析解读。全书分 5 篇 15 章，共约 600 个实例，内容有：开发环境、PHP 基础、Web 页面交互、文件操作、会话应用、图形图像处理、面向对象、MySQL 数据库与 PHP、MySQL 高级应用技术、PHP 操作 SQL Server 数据库、PHP 操作 Access 数据库、PHP 操作 Oracle 数据库、ADODB 类库、Smarty 模板、综合应用。配书光盘附带了实例的源程序和部分讲解视频。

《PHP 开发实例大全（基础卷）》既适合 PHP 程序员参考和查阅，也适合 PHP 初学者，如高校学生、软件开发培训学员及相关求职人员学习、练习、速查使用。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目（CIP）数据

PHP 开发实例大全．基础卷/软件开发技术联盟编著．—北京：清华大学出版社，2016
（软件工程师开发大系）
ISBN 978-7-302-38474-8

I. ①P…　II. ①软…　III. ①PHP 语言–程序设计　IV. ①TP312

中国版本图书馆 CIP 数据核字（2014）第 260802 号

责任编辑：赵洛育
封面设计：李志伟
版式设计：刘艳庆
责任校对：赵丽杰
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：203mm×260mm　　**印　　张：**57.5　　**字　　数：**1899 千字
（附光盘 1 张）
版　　次：2016 年 1 月第 1 版　　**印　　次：**2016 年 1 月第 1 次印刷
印　　数：1～3500
定　　价：128.00 元

产品编号：052243-01

前　言

Preface

特别说明：

《PHP 开发实例大全》分为基础卷（即本书）和提高卷两册。本书的前身是《PHP 开发实战 1200 例（第 I 卷）》。

编写目的

1. 方便程序员查阅

程序开发是一项艰辛的工作，挑灯夜战、加班加点是常有的事。在开发过程中，一个技术问题可能会占用几天甚至更长时间。如果有一本开发实例大全可供翻阅，从中找到相似的实例作参考，也许几分钟就可以解决问题。本书编写的主要目的就是方便程序员查阅、提高开发效率。

2. 通过分析大量源代码，达到快速学习之目的

本书提供了约 600 个开发实例及源代码，附有相应的注释、实例说明、关键技术、设计过程和秘笈心法，对实例中的源代码进行了比较透彻的解析。相信这种办法对激发学习兴趣、提高学习效率极有帮助。

3. 通过阅读大量源代码，达到提高熟练度之目的

俗话说“熟能生巧”，读者只有通过阅读、分析大量源代码，并亲自动手去做，才能够深刻理解、运用自如，进而提高编程熟练度，适应工作之需要。

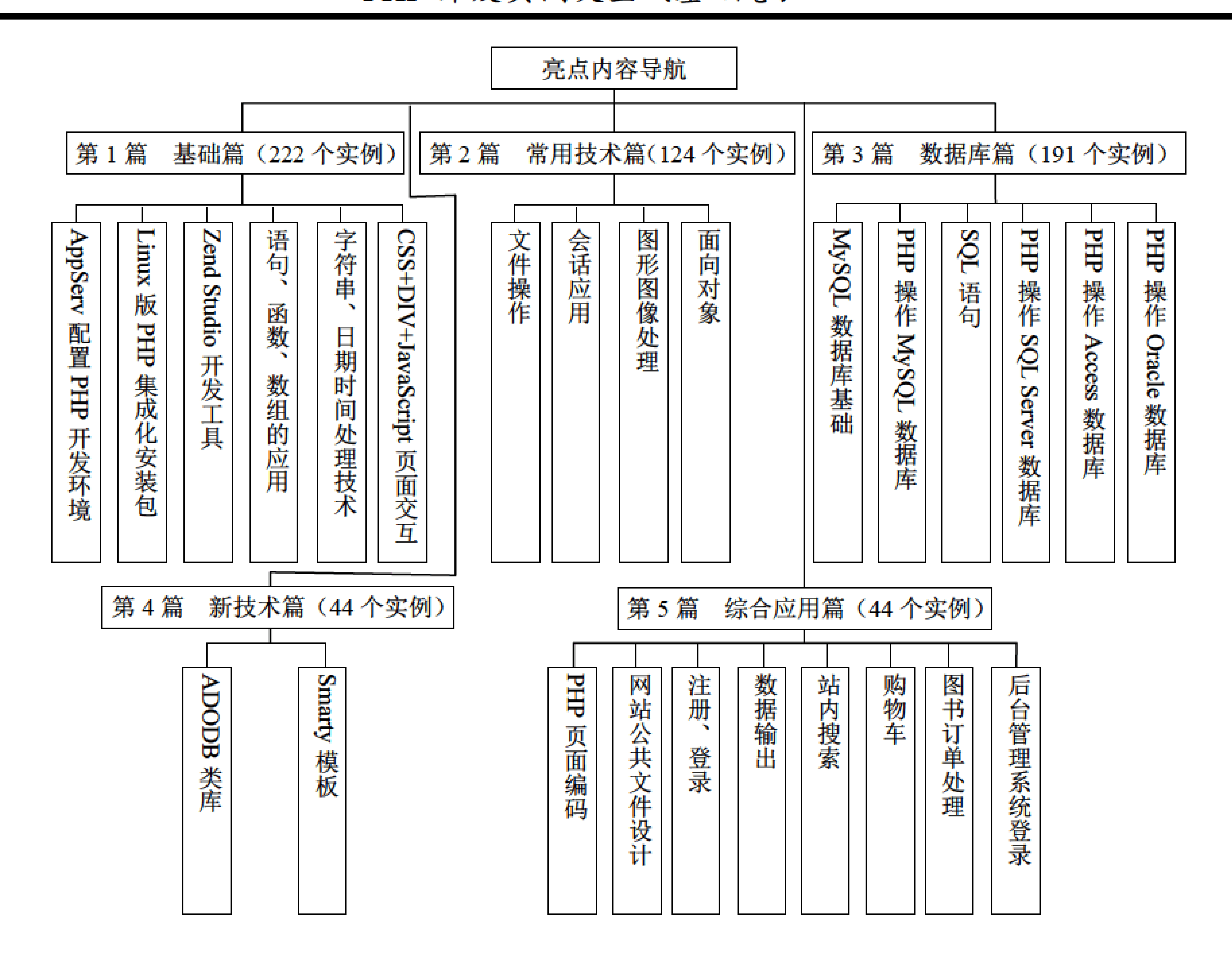
4. 实例源程序可以“拿来”就用，提高了效率

本书的很多实例，可以根据实际应用需求稍加改动，拿来就用，不必再去从头编写，从而节约时间，提高工作效率。

本书内容

全书分 5 篇 15 章，共约 600 个实例，内容有：开发环境、PHP 基础、Web 页面交互、文件操作、会话应用、图形图像处理、面向对象、MySQL 数据库与 PHP、MySQL 高级应用技术、PHP 操作 SQL Server 数据库、PHP 操作 Access 数据库、PHP 操作 Oracle 数据库、ADODB 类库、Smarty 模板、综合应用。书中所选实例均来源于一线开发人员的实际项目开发，囊括了开发中经常遇到和需要解决的热点、难点问题，使读者可以快速地解决开发中的难题，提高编程效率。本书知识结构如下图所示。

本书在讲解实例时采用统一的编排样式，多数实例由“实例说明”“关键技术”“设计过程”“秘笈心法”4 部分构成。其中，“实例说明”部分采用图文结合的方式介绍实例的功能和运行效果；“关键技术”部分介绍了实例使用的重点、难点技术；“设计过程”部分讲解了实例的详细开发过程；“秘笈心法”部分给出了与实例相关的技巧和经验总结。



本书特点

1. 实例极为丰富

本书精选了约 600 个实例，另外一册《PHP 开发实例大全（提高卷）》也精选了提高部分约 600 个实例，这样，两册图书总计约 1200 个实例，可以说是目前市场上实例最多、知识点最全面、内容最丰富的软件开发类图书，涵盖了编程中各个方面的应用。

2. 程序解释详尽

本书提供的实例及源代码，附有相应的注释、实例说明、关键技术、设计过程和秘笈心法。分析解释详尽，便于快速学习。

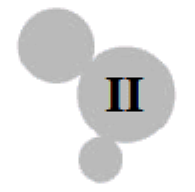
3. 实践实战性强

本书的实例及源代码很多来自现实开发中，光盘中绝大多数实例给出了全部源代码，读者可以直接调用、研读、练习。

关于光盘

1. 实例学习注意事项

读者在按照本书学习、练习的过程中，可以从光盘中复制源代码，修改时注意去掉源码文件的只读属性。**有些实例需要使用相应的数据库或第三方资源，在使用前需要进行相应配置，具体步骤请参考书中或者光盘中的配置说明。**



2. 实例源代码及视频位置

本书光盘提供了实例的源代码，位置在光盘中的“MR\章号\实例序号”文件夹下，例如，“MR\04\226”表示实例 226，位于第 4 章。部分实例提供的视频讲解，也可根据以上方式查找。**由于有些实例源代码较长，限于篇幅，图书中只给出了关键代码，完整代码放置在光盘中。**

3. 视频使用说明

本书提供了部分实例的视频讲解，在目录中标题前边有视频图标的实例，即表示在光盘中有视频讲解。视频采用 EXE 文件格式，无须使用播放器，双击就可以直接播放。

读者对象

PHP 程序员，PHP 初学者，如高校大学生、求职人员、培训机构学员等。

本书服务

如果您使用本书的过程中遇到问题，可以通过如下方式与我们联系。

☑ 服务 QQ：4006751066
☑ 服务网站：http://www.mingribook.com

本书作者

本书由软件开发技术联盟组织编写，参与编写的程序员有赛奎春、王小科、王国辉、王占龙、高春艳、张鑫、杨丽、辛洪郁、周佳星、申小琦、张宝华、葛忠月、王雪、李贺、吕艳妃、王喜平、张领、杨贵发、李根福、刘志铭、宋禹蒙、刘丽艳、刘莉莉、王雨竹、刘红艳、隋光宇、郭鑫、崔佳音、张金辉、王敬洁、宋晶、刘佳、陈英、张磊、张世辉、高茹、陈威、张彦国、高飞、李严。在此一并致谢！

编　者

目　录

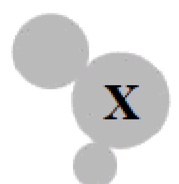
Contents

第 1 篇　基础篇

第 2 篇　常用技术篇

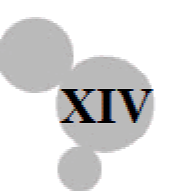


第 3 篇 数据库篇



第 4 篇　新技术篇

第 5 篇　综合应用篇





基础篇

- ⏭ 第 1 章　开发环境
- ⏭ 第 2 章　PHP 基础
- ⏭ 第 3 章　Web 页面交互

第 1 章

开发环境

- AppServ——PHP 集成化安装包
- XAMPP——PHP 集成化安装包
- IIS+PHP+MySQL——独立搭建 PHP 开发环境
- LAMP——独立配置 PHP 开发环境
- XAMPP——Linux 版 PHP 集成化安装包
- Dreamweaver 开发工具
- Zend Studio 开发工具

1.1　AppServ——PHP 集成化安装包

AppServ 将 Apache、PHP、MySQL 和 phpMyAdmin 等服务器软件和工具安装配置完成后进行打包处理，开发人员只要到网站上下载该软件然后安装，即可完成 PHP 开发环境的快速搭建，非常适合初学者使用。

实例 001	通过 AppServ 配置 PHP 开发环境 光盘位置：光盘\MR\01\001	初级 趣味指数：★★★★

实例说明

在本实例中，讲解如何通过 AppServ 配置 PHP 的开发环境，配置完成后将出现如图 1.1 所示对话框。



图 1.1　通过 AppServ 配置 PHP 开发环境

关键技术

通过 AppServ 配置 PHP 开发环境时，首先从互联网上下载 AppServ 的安装文件，下载地址为 http://www.AppServnetwork.com/。

注意： 在使用 AppServ 搭建 PHP 开发环境时，必须确保在系统中没有安装 Apache、PHP 和 MySQL，否则要先将这些软件卸载，然后应用 AppServ。

在软件的安装过程中，要重点注意下面几个步骤：

☑　设置安装路径。
☑　选择安装软件。
☑　设置端口号。
☑　设置数据库密码。
☑　设置数据库编码格式。

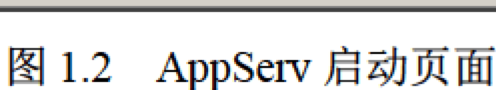
设计过程

下面对应用 AppServ 集成化安装包搭建 PHP 开发环境的具体操作步骤进行讲解。

（1）双击 AppServ-win32-2.5.10.exe 文件，打开如图 1.2 所示的 AppServ 启动对话框。

（2）单击图 1.2 中的 Next 按钮，进入如图 1.3 所示的 AppServ 安装协议对话框。




图 1.2　AppServ 启动页面

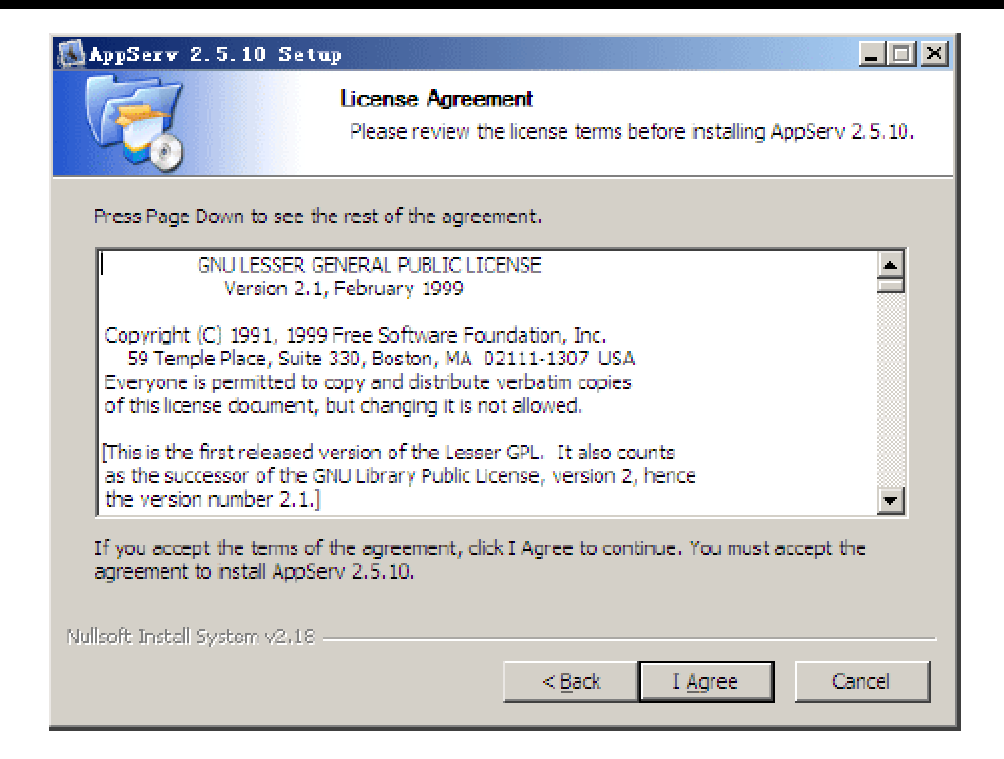


图 1.3　AppServ 安装协议

（3）单击 I Agree 按钮，进入如图 1.4 所示的对话框，在其中设置 AppServ 的安装路径（默认安装路径一般为 C:\AppServ），AppServ 安装完成后，Apache、MySQL、PHP 都将以子目录的形式存储到该目录下。

（4）单击 Next 按钮，进入如图 1.5 所示的对话框，在其中选择要安装的程序和组件（默认为全选）。

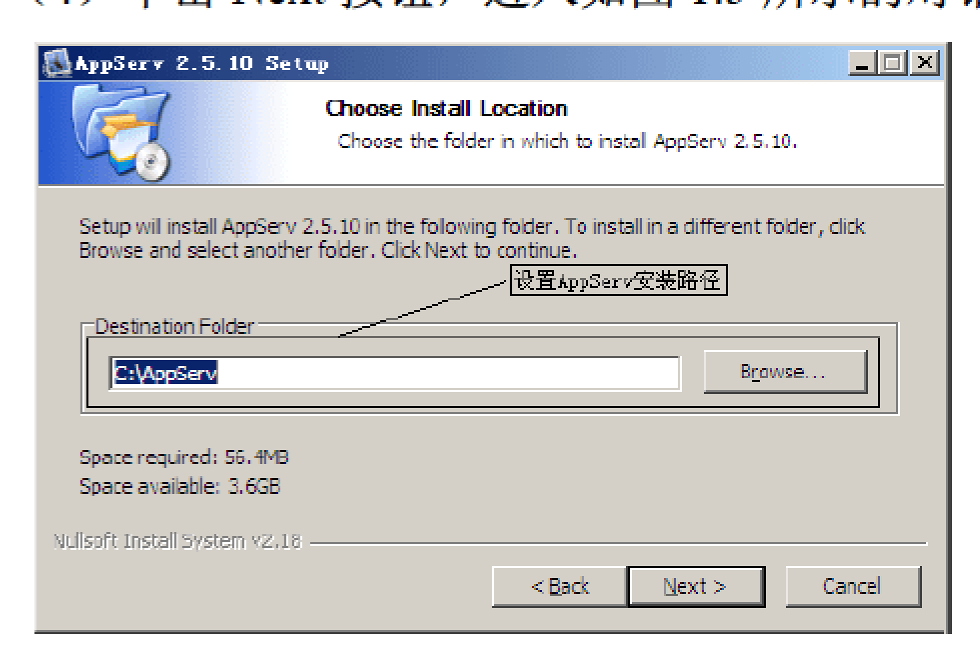


图 1.4　AppServ 安装路径选择

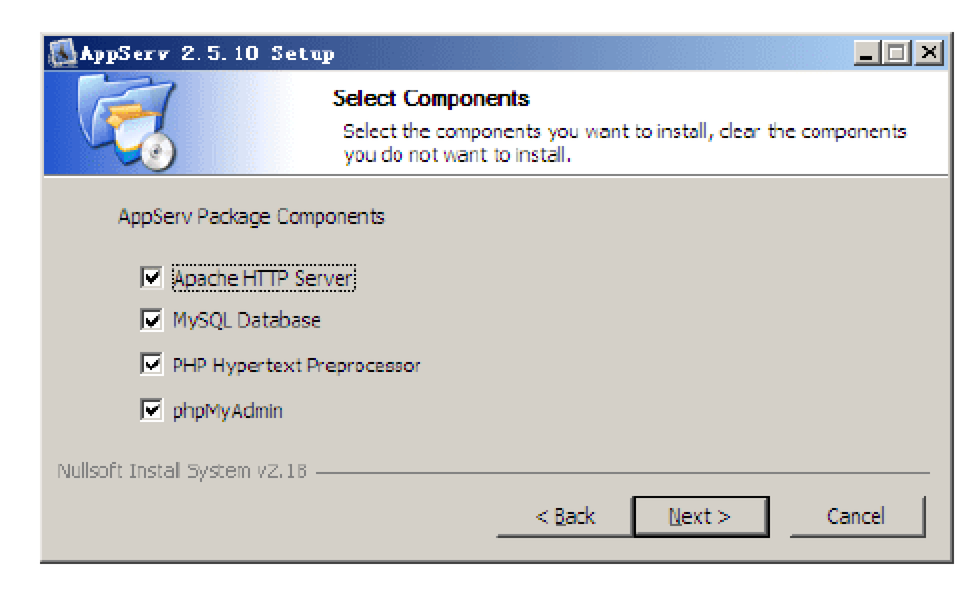


图 1.5　AppServ 安装选项

技巧： 在第（4）步中，如果本机已经安装 MySQL 数据库，那么可以不选中 MySQL Database 复选框，仍使用本机已经存在的 MySQL 数据库。

（5）单击图 1.5 中的 Next 按钮，进入如图 1.6 所示的对话框，在其中填写计算机名称、添加邮箱地址、设置 Apache 的端口号（默认为 80 端口）。

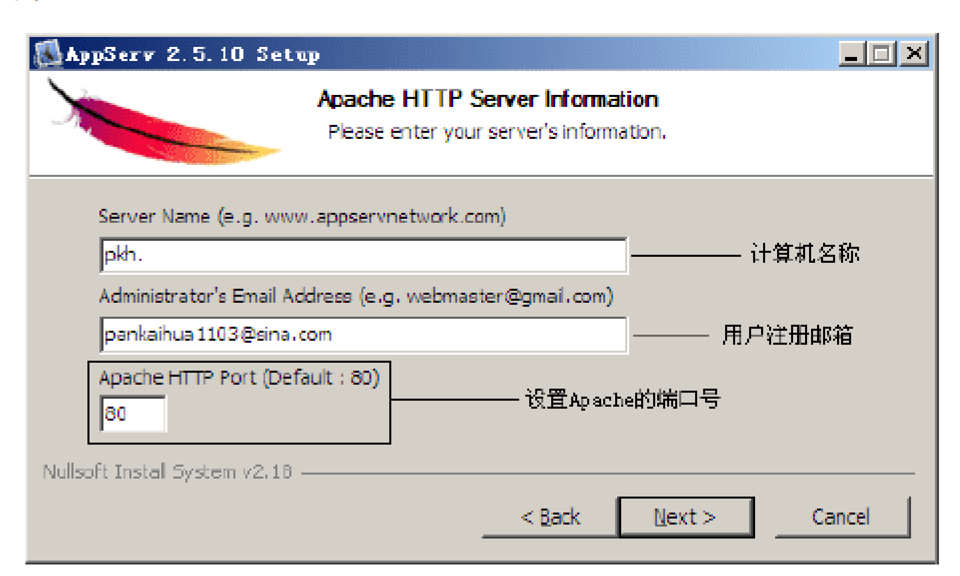
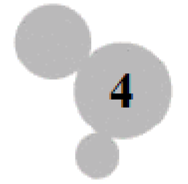


图 1.6　Apache 端口号设置

注意： Apache 服务器端口号的设置，直接关系到 Apache 服务器是否能够正常启动。如果本机中的 80 端口被 IIS 或者迅雷占用，那么这里就需要修改 Apache 的端口号，或者修改 IIS 和迅雷的端口号，这样才能完成 Apache 服务器的配置。如果出现端口冲突，那么将导致安装失败，Apache 服务不能启动。



（6）单击 Next 按钮，进入如图 1.7 所示的对话框，在其中设置 MySQL 数据库 root 用户的登录密码及字符集。

说明： MySQL 数据库的字符集，可以设置为 UTF-8、GBK 或者 GB2312 格式。这里将字符集设置为 GB2312 Simplified Chinese，表示 MySQL 数据库的字符集将采用简体中文形式。

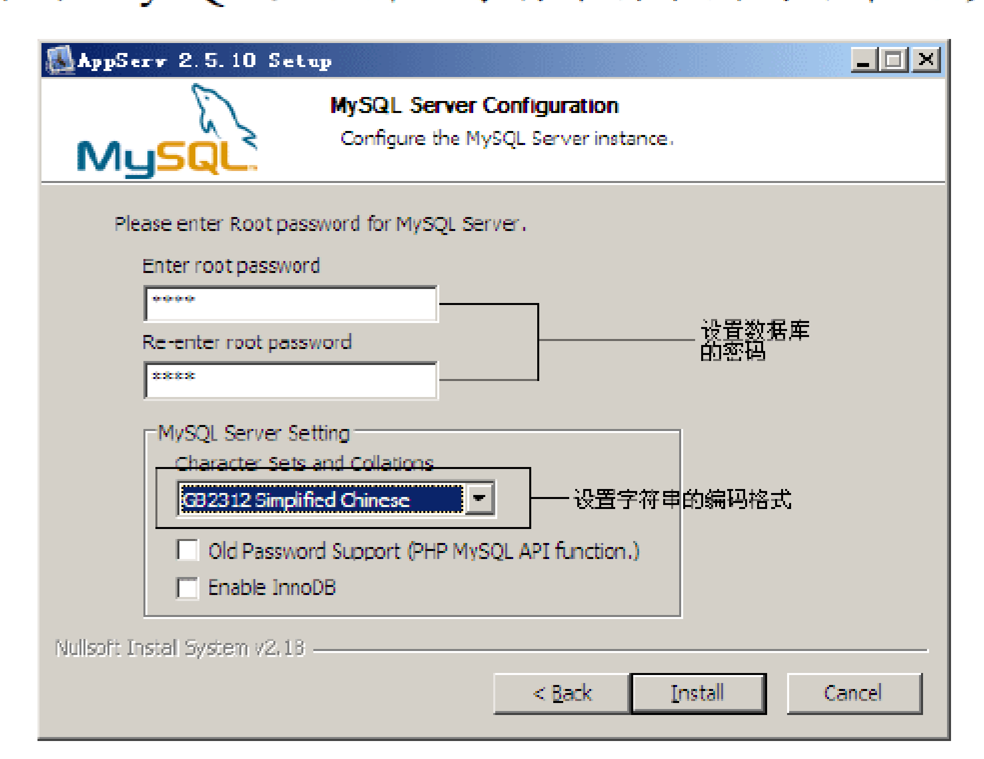


图 1.7　设置 MySQL 数据库 root 用户的密码和字符集

技巧： 对于在图 1.7 中设置的 MySQL 数据库 root 用户的密码必须牢记，因为程序在连接数据库时必须使用这个密码。建议读者将这个密码设置为 111，因为这是在开发本书中的程序时所使用的数据库密码。如此，在运行本书中的数据库程序时，就不需要修改密码，否则要修改连接数据库的密码。如果忘记安装时设置的密码，最直接有效的解决方式是重新安装 AppServ。

（7）单击 Install 按钮开始安装，如图 1.8 所示。

（8）安装完成后可以在开始菜单的 AppServ 相关操作列表中启动 Apache 及 MySQL 服务，如图 1.9 所示。

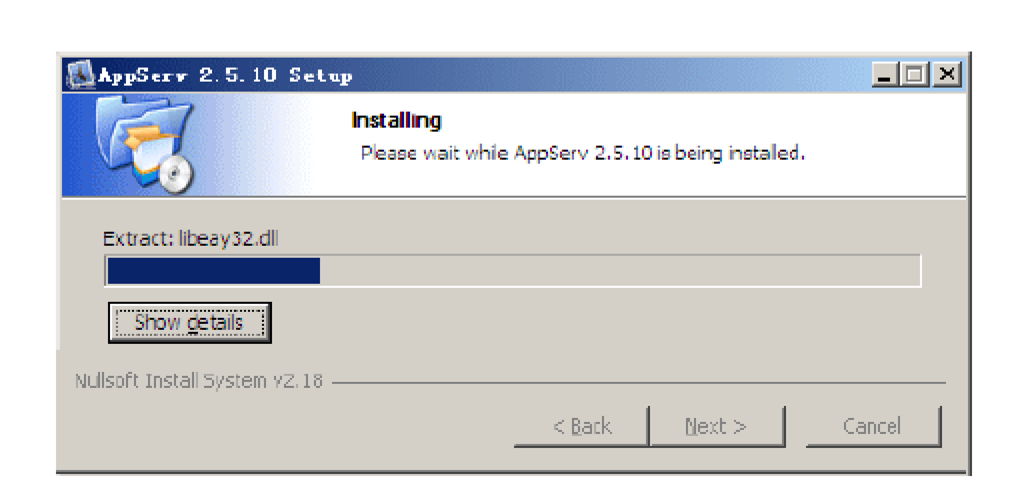


图 1.8　AppServ 安装页面

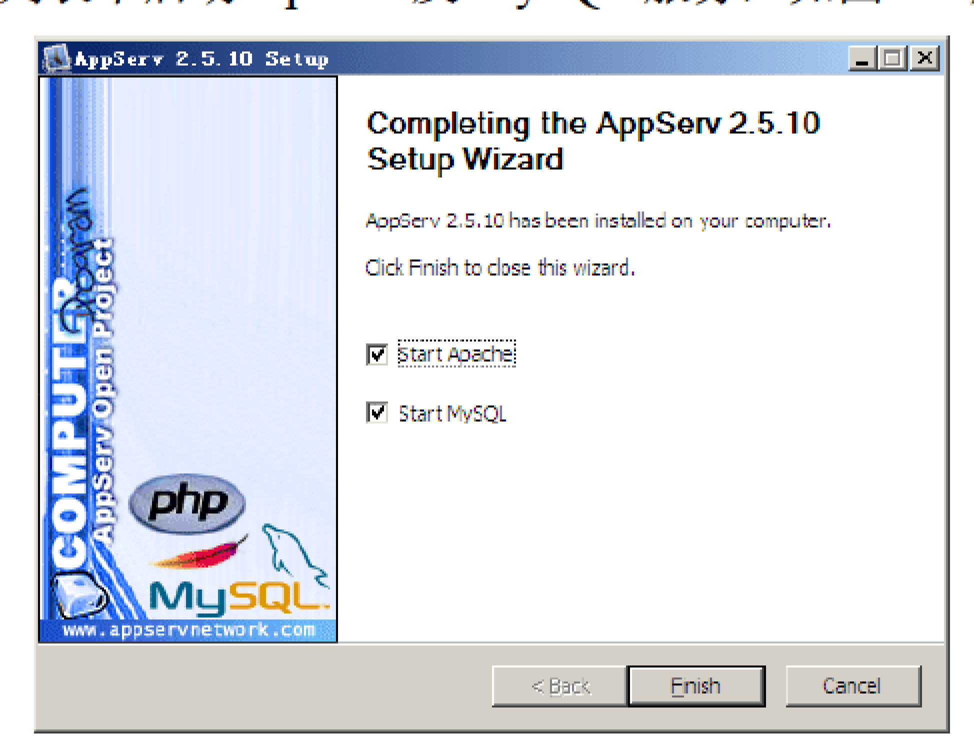


图 1.9　AppServ 安装完成页面

秘笈心法

心法领悟 001：Apache 的配置文件。

Apache 的配置文件存储于 C:\AppServ\Apache2.2\conf\httpd.conf。通过 Apache 配置文件可以设置 Apache 的端口号、Apache 服务器的根目录或者控制 Apache 是否支持伪静态。

（1）设置 Apache 的端口号：

```
Listen 80
```

（2）设置 Apache 服务器的根目录：

```
DocumentRoot "C:/AppServ/www"
```

（3）设置对伪静态的支持：

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

将该项前面的“#”去掉，可启动该项。查找 httpd.conf 文件，找到其中的 AllowOverride 项，将其值都修改为 All。

完成对 Apache 的修改后，必须保存文件，并重新启动 Apache 服务器才能够生效。

实例 002 AppServ 应用技巧

光盘位置：光盘\MR\01\002

中级

趣味指数：★★★☆

实例说明

安装 AppServ 后，可以对其进行测试。在 IE 浏览器的地址栏中输入 hppt://127.0.0.1，如果输出如图 1.10 所示的页面，则说明 PHP 开发环境配置成功。

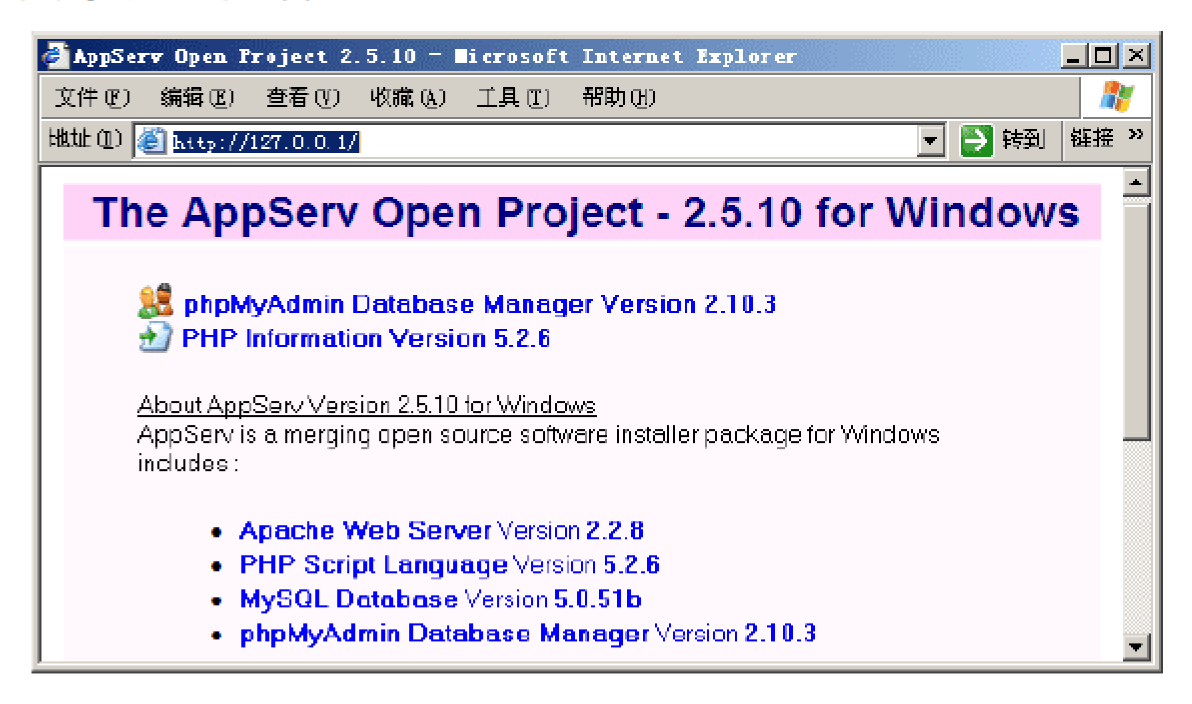


图 1.10 PHP 开发环境配置成功

关键技术

AppServ 安装完成后，整个目录默认安装在 C:\AppServ 下，此目录下包含 4 个子目录，如图 1.11 所示，用户可以将所有程序文件存储到 www 目录下。

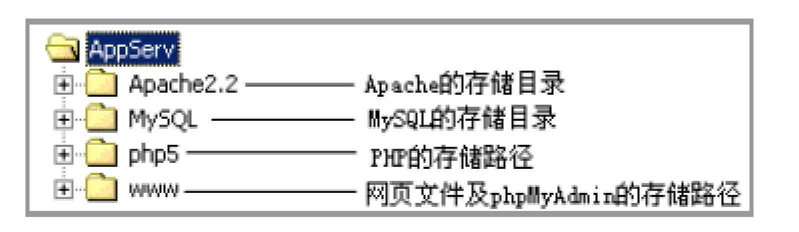


图 1.11 AppServ 目录结构

在 Apache2.2\conf\目录下，有一个 httpd.conf 文件，它是 Apache 服务器的配置文件，在这个文件中可以修改 Apache 服务器的端口号、根目录等，所有有关 Apache 服务器的配置都在这个文件中完成。

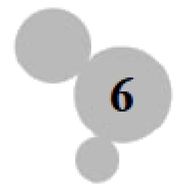
在 MySQL 目录下，有一个 my.ini 文件，它是 MySQL 服务器的配置文件，存储 MySQL 的配置信息。

在 MySQL\data 目录下存储的是数据库文件，所有程序使用的数据库都存储在这个文件夹下。

说明：如果 MySQL 服务器的版本是 5.1，那么数据库文件夹 data 的默认存储位置是 C:\Documents and Settings\All Users\Application Data\MySQL\MySQL Server 5.1\Data\。

在 php5\ext 文件夹下存储的是 PHP 内置的函数类库，以.dll 的格式存储，而 PHP 的配置文件 php.ini 存储在本机系统盘的 Windows 文件夹下。

www 目录是程序运行的根目录，也就是说我们要运行的所有程序都必须存储在这个目录下。phpMyAdmin



图形化管理工具默认存储在此目录下。

设计过程

（1）www 文件夹存储网站的运行文件，其默认的内容如图 1.12 所示。

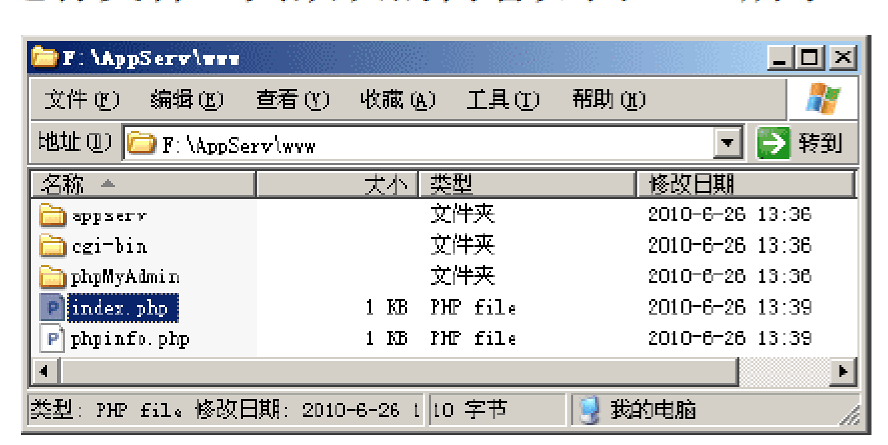


图 1.12　www 文件夹的默认内容

（2）其中，index.php 是默认访问 127.0.0.1 时浏览的内容；而 phpinfo.php 存储的是 PHP 的配置信息和支持的模块。phpinfo.php 的内容如图 1.13 所示。

PHP Version 5.3.1　php

System	Windows NT PKH 5.2 build 3790 (Windows Server 2003 Standard Edition) i586
Build Date	Nov 20 2009 17:20:57
Compiler	MSVC6 (Visual C++ 6.0)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled

图 1.13　phpinfo.php 的内容

（3）其中的 phpMyAdmin 文件夹存储的是 phpMyAdmin 图形化管理工具。

秘笈心法

心法领悟 002：AppServ 的相关说明。

（1）如果是应用 AppServ 配置的 PHP 开发环境，那么在 IE 浏览器中直接输入 127.0.0.1 将输出如图 1.10 所示的页面，因为在 www 文件夹下有一个默认的 index.php 文件；如果要运行 www 文件夹下的其他文件，那么必须在 127.0.0.1 后指定具体的文件夹或者文件，例如，输入 127.0.0.1/text.php 或者 127.0.0.1/mr/。

（2）Apache 的启动和停止。

单击“开始”按钮，选择“程序”/AppServ/Control Server by Service 命令，在弹出的菜单中完成 Apache、MySQL 服务器的启动和停止操作。

单击“开始”按钮，选择“设置”/“控制面板”/“管理工具”/“服务”命令，找到 Apache 2.2，右击，在弹出的快捷菜单中完成 Apache 的启动、停止或者重新启动操作。

完成对 php.ini 文件的修改后，同样要保存文件，重新启动 Apache 服务器后，修改才能够生效。

实例 003　第 1 个 PHP 程序

光盘位置：光盘\MR\01\003

初级　趣味指数：★★★☆

实例说明

AppServ 安装、调试成功后，开发第 1 个 PHP 程序，输出系统的当前时间，运行结果如图 1.14 所示。



图 1.14　第 1 个 PHP 程序

关键技术

（1）应用 PHP 中的 date()函数，输出系统的当前时间，设置时间的格式为年-月-日 时:分:秒（Y-m-d H:i:s）。

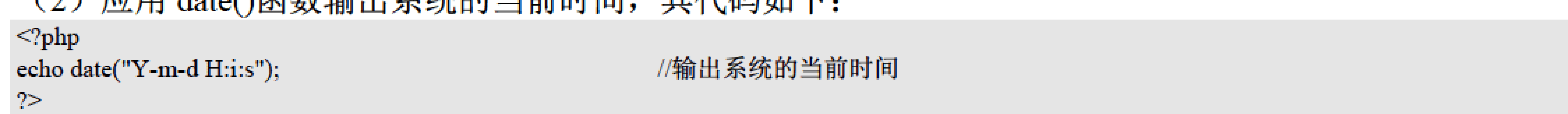
（2）将编写的 index.php 文件存储于 www 文件夹下的 MR\01\003 目录下。

在运行本实例时，会发现程序输出的时间与系统的当前时间相差了 8 个小时，至于为什么会出现此种情况，将在本实例的秘笈心法中讲解。

设计过程

（1）通过 Dreamweaver 创建一个 index.php 脚本文件，存储于 MR/01/003/文件夹下。

（2）应用 date()函数输出系统的当前时间，其代码如下：



```
<?php
echo date("Y-m-d H:i:s");                                    //输出系统的当前时间
?>
```

（3）在 IE 浏览器的地址栏中输入 http://127.0.0.1/mr/01/003/，运行结果如图 1.14 所示。

秘笈心法

心法领悟 003：对于系统时间的相关说明。

程序输出时间与系统当前时间出现偏差的关键是 php.ini 文件的设置。在 php.ini 文件中，默认设置的是标准的格林威治时间，即采用的是英国伦敦时间，而当地时间是北京时间，所以会出现时间的偏差。

PHP 的配置文件存储于 C:\Windows\php.ini 文件中。在 php.ini 文件中，定位到[date]下的“;date.timezone =”选项，去掉前面的分号，并设置它的值为当地所在时区使用的时间，即可统一程序输出的时间与系统当前的时间。

```
;date.timezone =
```

例如，如果当地所在时区为东八区，那么就可以设置 date.timezone 的值为 PRC、Asia/Hong_Kong、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐）等，这些都是东八区的时间。

1.2　XAMPP——PHP 集成化安装包

XAMPP（Apache+MySQL+PHP+Perl）是一个功能强大的建站集成软件包，可以在 Windows、Linux、Solaris 这 3 种操作系统下安装使用，支持多种语言，包括英文、简体中文、繁体中文、韩文、俄文、日文等。

实例 004	通过 XAMPP 配置 PHP 开发环境 光盘位置：光盘\MR\01\004	初级 趣味指数：★★★☆

实例说明

XAMPP 是一个易于安装且包含 MySQL、PHP 和 Perl 的 Apache 发行版。安装成功后的运行效果如图 1.15 所示。

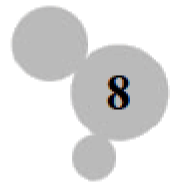




图 1.15　XAMPP 安装成功

关键技术

目前 XAMPP 的 Windows 版本是 1.7.3，包括如下内容：

- ☑ Apache 2.2.14 (IPv6 enabled)
- ☑ MySQL 5.1.41 + PBXT engine
- ☑ PHP 5.3.1
- ☑ OpenSSL 0.9.8l
- ☑ phpMyAdmin 3.2.4
- ☑ XAMPP Control Panel 2.5.8
- ☑ XAMPP CLI Bundle 1.6
- ☑ Webalizer 2.21-02
- ☑ Mercury Mail Transport System v4.72
- ☑ msmtp 1.4.19
- ☑ FileZilla FTP Server 0.9.33
- ☑ SQLite 2.8.17
- ☑ SQLite 3.6.20
- ☑ ADOdb 5.10
- ☑ eAccelerator 0.9.6-rc1
- ☑ Xdebug 2.0.6-dev
- ☑ Ming 0.4.3

适用于 Windows 2000/XP/2003/Vista/7。

在 Windows 操作系统下，XAMPP 有两种形式：

- ☑ 自解压 RAR 包——简单且安全
- ☑ ZIP 压缩包——XAMPP 的普通 ZIP 压缩档案

设计过程

XAMPP 1.7.3 版的具体安装步骤如下：

（1）运行.exe 安装文件，弹出如图 1.16 所示对话框，在这里设置 XAMPP 的安装路径。

（2）安装路径设置完成后，单击 Install 按钮，执行 XAMPP 的安装，如图 1.17 所示。

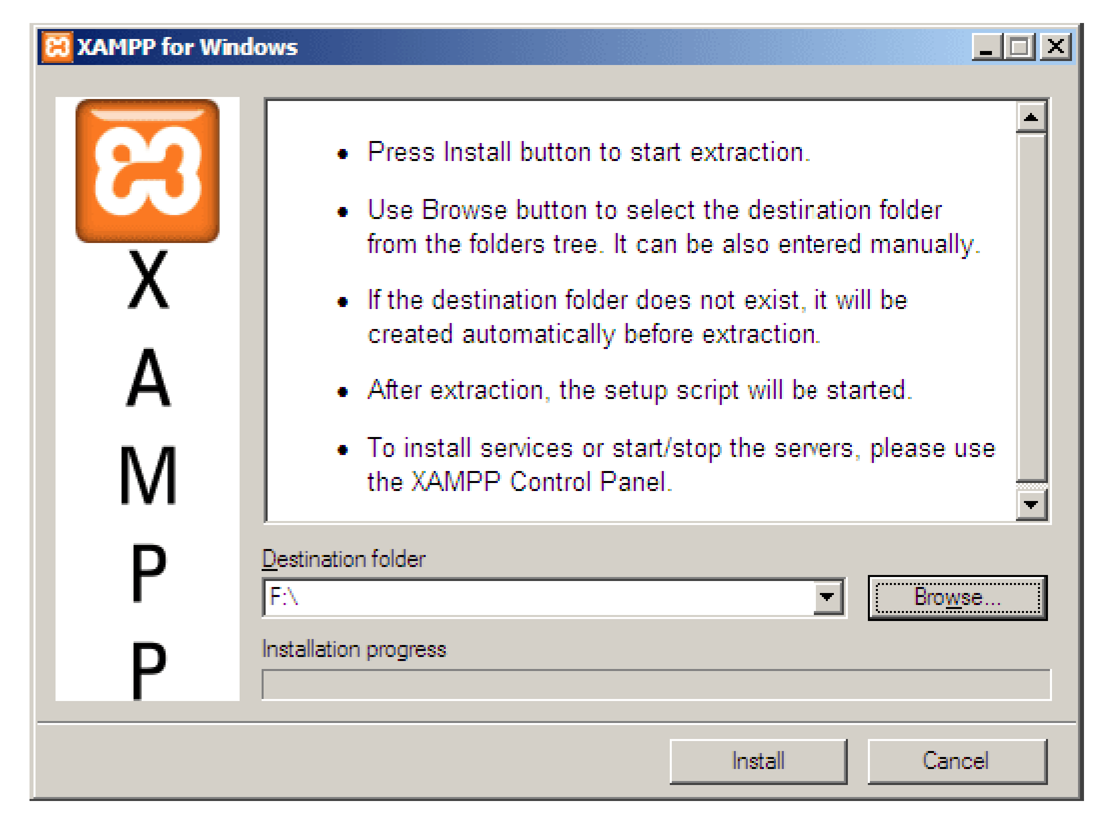


图 1.16　设置 XAMPP 的安装路径

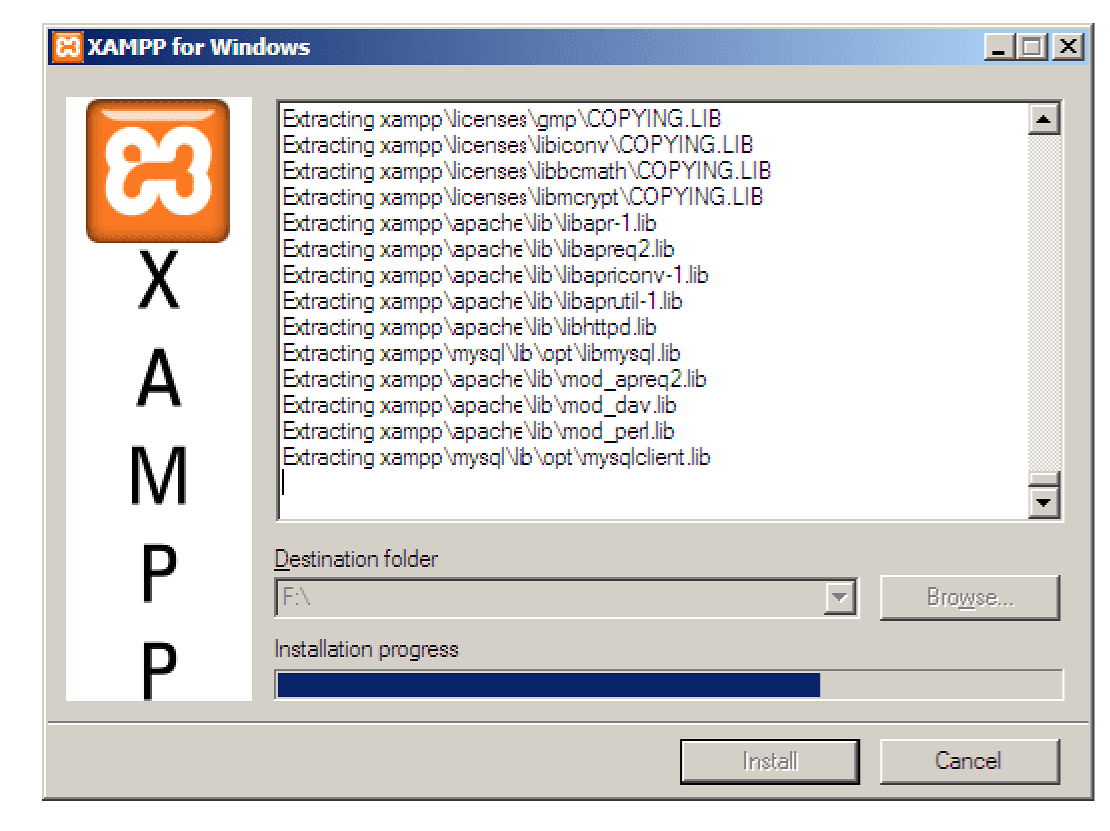


图 1.17　安装 XAMPP

（3）XAMPP 安装成功后，将弹出如图 1.18 所示的操作界面，在其中对 XAMPP 进行设置，包括是否创建桌面快捷方式、是否在开机时启动 XAMPP 等。

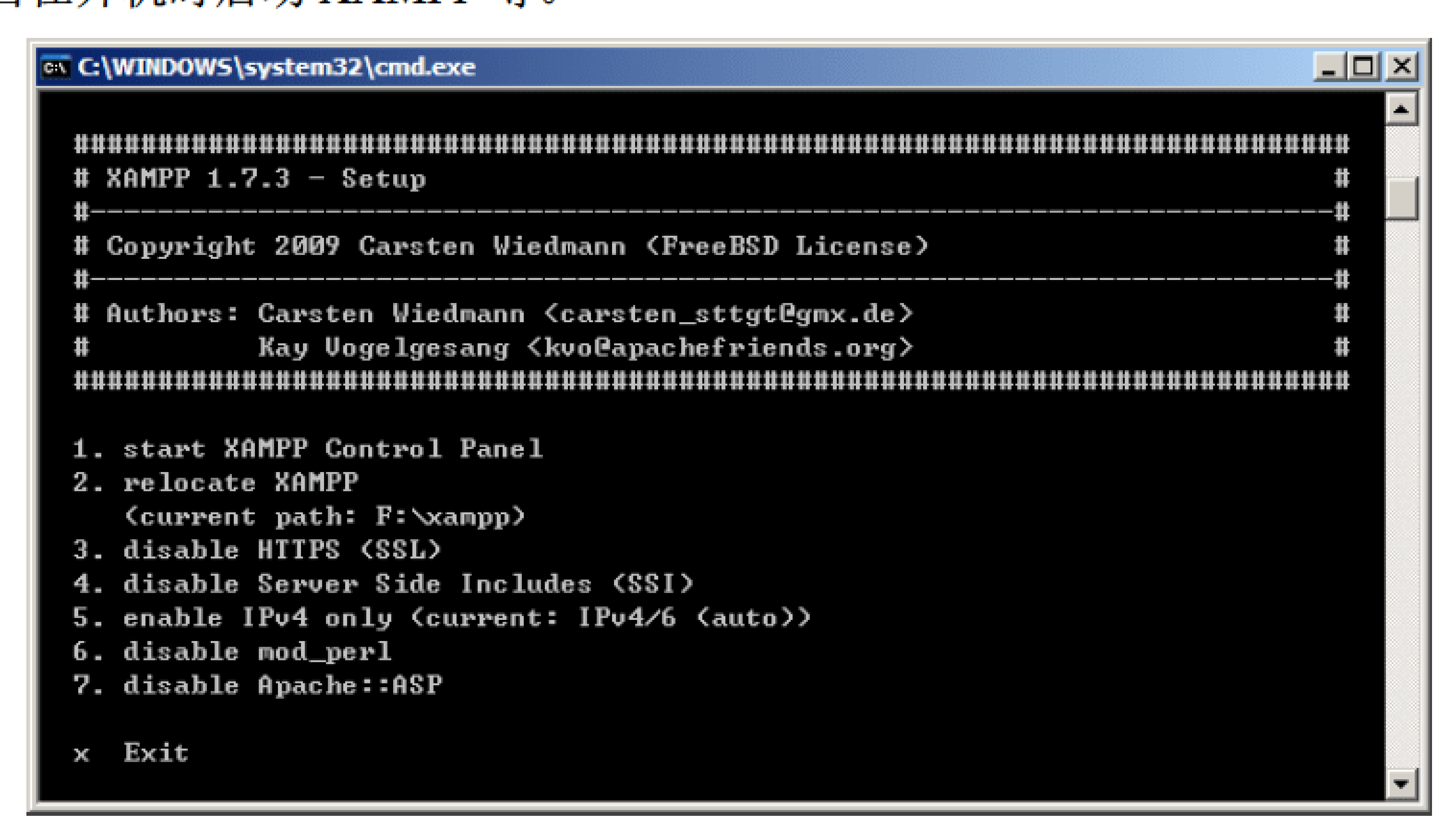


图 1.18　设置 XAMPP

至此，XAMPP 安装、配置完毕。

秘笈心法

心法领悟 004：XAMPP 中的配置文件。

XAMPP 中各种配置文件的存储位置如下。

☑　Apache 基本配置：.\xampp\apache\conf\httpd.conf。

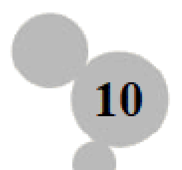
☑　Apache SSL：.\xampp\apache\conf\ssl.conf。

☑　Apache Perl（仅限插件）：.\xampp\apache\conf\perl.conf。

☑　PHP：.\xampp\php\php.ini。

☑　MySQL：.\xampp\mysql\bin\my.ini。

☑　phpMyAdmin：.\xampp\phpMyAdmin\config.inc.php。



☑　FileZilla FTP 服务器：.\xampp\FileZillaFTP\FileZilla。

实例 005　测试 XAMPP 是否安装成功

光盘位置：光盘\MR\01\005

初级

趣味指数：★★★☆

实例说明

成功安装 XAMPP 后，启动 Apache 服务器，在浏览器的地址栏中输入 http://localhost 或 http://127.0.0.1，即可检测所有的 XAMPP 样例和工具，运行结果如图 1.19 所示。

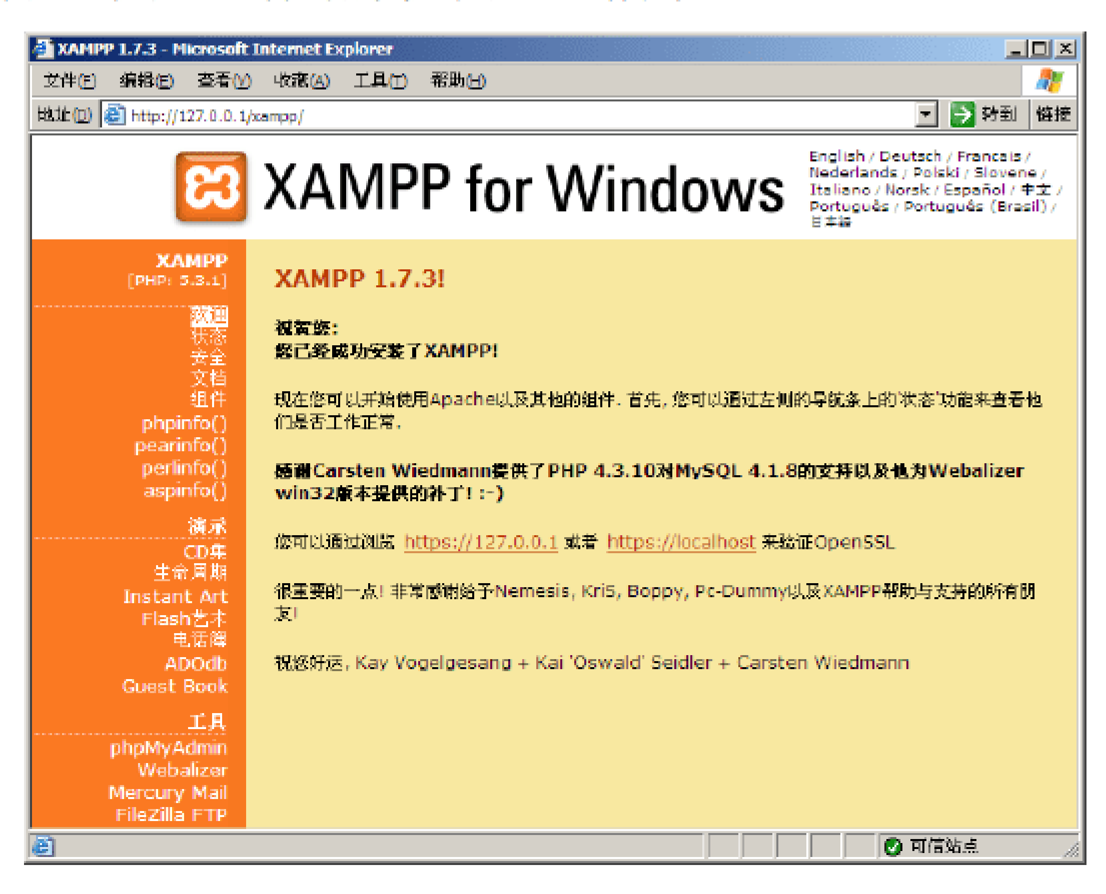


图 1.19　检测安装是否成功

关键技术

XAMPP 中默认将所有网络文档存储于 htdocs 主文件夹中（.\xampp\htdocs）。默认 htdocs 主文件夹下的内容如图 1.20 所示。



图 1.20　XAMPP 的默认网络文件

读者可以在浏览器的地址栏中输入 http://localhost/来检测 XAMPP 是否安装成功。

设计过程

（1）XAMPP 文件夹存储的是 XAMPP 的各种配置信息，当直接运行 index.php 文件时，调用的就是 XAMPP 文件夹下的 index.php 文件，输出如图 1.20 所示的内容。

（2）在 index.html 文件中，输出一个简单的测试信息，如图 1.21 所示。



图 1.21　通过 HTML 测试 PHP 是否安装成功

（3）在 index.php 文件中，通过 header()函数跳转到 xampp 文件夹下的 index.php 文件，输出 XAMPP 的信息。

秘笈心法

心法领悟 005：启动、停止和测试 XAMPP。

通过文件\xampp\xampp-control.exe 控制 XAMPP 的启动、停止和测试。

实例 006　XAMPP 应用技巧

光盘位置：光盘\MR\01\006

初级

趣味指数：★★★☆

实例说明

XAMPP 只为开发者服务，切记不要将其应用于生产环境，因为对于生产环境而言，使用 XAMPP 可能会带来灾难性的打击。

因为 MySQL 管理员（root）未设置密码，用户可以直接通过 phpMyAdmin 访问 MySQL 服务器，修改数据库中的数据，或者通过网络访问，运行结果如图 1.22 所示。



图 1.22　直接访问 MySQL 数据库

关键技术

要确保 MySQL 数据库的安全，必须为 MySQL 管理员（root）设置密码，其方法如下：

以 root 用户的身份登录 phpMyAdmin，在 phpMyAdmin 的权限中设置 root 用户的密码。操作界面如图 1.23 所示。

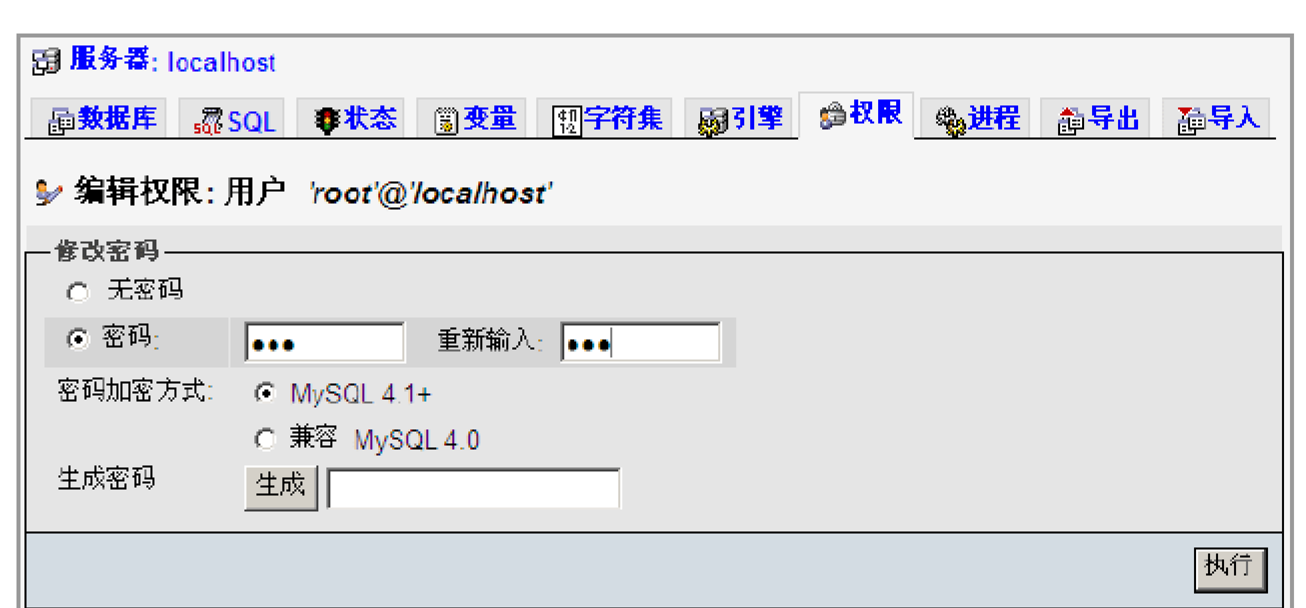
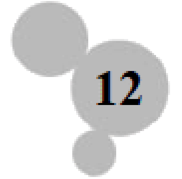


图 1.23　修改 MySQL 用户密码

说明：更改 root 的密码之后，还要修改 phpMyAdmin 中的相关信息，具体方法请参看设计过程。



设计过程

在 phpMyAdmin 的权限中完成 root 用户的密码设置后，在...\xampp\phpmyadmin\下搜索 config.inc.php 文件，定位到如下位置：

```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'http';                //设置登录方式
$cfg['Servers'][$i]['user'] = 'root';                     //设置用户名
$cfg['Servers'][$i]['password'] = "111;                   //设置密码
$cfg['Servers'][$i]['AllowNoPassword'] = true;
```

设置 auth_type 的值为“http”；设置 password 的值为“111”，这样登录 phpMyAdmin 时，就必须要输入密码，否则无法登录。

秘笈心法

心法领悟 006：修改 MySQL 默认字符集。

修改 MySQL 默认字符集为 utf8，定位到 my.ini 文件，需添加如下内容：

```
[client]
default_character_set=utf8
[mysqld]
# default-character-set=utf8
character-set-server = utf8
collation-server = utf8_general_ci
[mysql]
default_character_set=utf8
```

实例 007	第 2 个 PHP 程序 光盘位置：光盘\MR\01\007	初级 趣味指数：★★★★☆

实例说明

既然 XAMPP 只为开发者服务，那么我们就应用 XAMPP 开发一个 PHP 程序，熟悉一下 XAMPP 的应用。本实例简单地输出一段文字，其运行结果如图 1.24 所示。

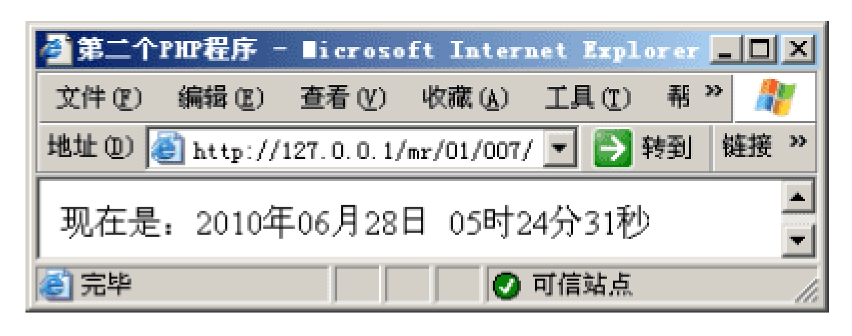


图 1.24　第 2 个 PHP 程序

关键技术

（1）确定服务器的运行文件夹，编写 PHP 脚本文件，将其存储于 XAMPP 指定的运行文件夹下。

（2）应用 PHP 的 date()函数输出系统的当前时间，时间的格式是“年 月 日 时 分 秒”。

设计过程

（1）新建一个记事本文件，编写 PHP 脚本，通过 date()函数输出系统的当前时间，其代码如下：

```
<?php
echo "现在是：".date("Y 年 m 月 d 日  H 时 i 分 s 秒");
?>
```

（2）保存这个记事本文件，将其存储于 xampp\htdocs\MR\01\007（这是笔者在本机中定义的实例存储文件夹）。

秘笈心法

心法领悟 007：配置默认的开始页面。

如果在浏览器地址栏中输入 http://localhost/xampp/，Apache 服务器会自动返回一个默认的开始页面，Apache 会自动寻找已存在的类似 index.php 的索引页。httpd.conf 中的 DirectoryIndex 指令负责这个功能。在这里可以定义默认开始页面的名字和协议。

XAMPP 在默认状态下，DirectoryIndex 指令的列表如下：

```
index.php index.php4 index.php3 index.cgi index.pl index.html index.htm index.html.var index.phtml
```

1.3 IIS+PHP+MySQL——独立搭建 PHP 开发环境

Internet 信息服务器即 Internet Information Server，缩写为 IIS，是 Microsoft 的 Web 服务器。它集成于 Windows NT Server 之中，方便易用，为 Web 应用程序提供了功能强大的运行平台。

本书使用 IIS 6.0，在 Windows 2003 操作系统下完成 PHP 环境的搭建，所使用的软件如下。

☑ PHP 软件：php-5.3.1-nts-Win32-VC9-x86。

☑ MySQL 数据库：mysql-essential-5.1.39-win32.msi。

说明：上述版本是笔者所使用的，读者可以根据自己的实际情况进行调整。如果是 Windows 7 操作系统，则使用 IIS 7.0，如果是 Windows XP 系统，则使用 IIS 5.0。

实例 008	安装 PHP 光盘位置：光盘\MR\01\008	初级 趣味指数：★★★★☆

实例说明

PHP 5.3.1 是使用较多的版本，读者可以到 PHP 的官方网站下载，下载地址为 http://www.php.net/ downloads. php。本实例讲解如何选择、安装 PHP 5.3.1。

关键技术

（1）版本的区分。PHP 推出的 PHP 5.3.1 有多个版本，包括 VC6、X86 和 VC9、X86。

☑ VC 6：legacy Visual Studio 6 compiler，是使用此编译器编译的。

☑ VC 9：the Visual Studio 2008 compiler，是用微软的 VS 编辑器编译的。

（2）版本的选择。

☑ 如果在 Windows 下使用 Apache+PHP，选择 VC 6 版本；

☑ 如果在 Windows 下使用 IIS+PHP，选择 VC 9 版本。

（3）Non Thread Safe 和 Thread Safe。Non Thread Safe 是非线程安全；Thread Safe 是线程安全。官方不建议将 Non Thread Safe 应用于生产环境，所以我们选择 Thread Safe 版本的 PHP。

设计过程

下面讲解 php-5.3.1-nts-Win32-VC9-x86 的配置方法。

（1）将下载的文件 php-5.3.1-nts-Win32-VC9-x86.zip 解压，重新命名为 php，并将其复制到指定的目录下。



例如 C:\php、F:\php 等。这里将 php 放置在 F:\PkhiisPHP 目录下。

（2）将 F:\PkhiisPHP\php 目录下的 libMySQL.dll 文件复制到 C:\Windows\system32\（如果是 Windows 2000，则为 C:\WINNT\system32\）目录下。

（3）将 F:\PkhiisPHP\php 目录下的 php.ini-development 重命名为 php.ini，如图 1.25 所示，并将其复制到系统盘的 Windows 文件夹下（例如，若 C 盘为系统盘，则是 C:\Windows）。

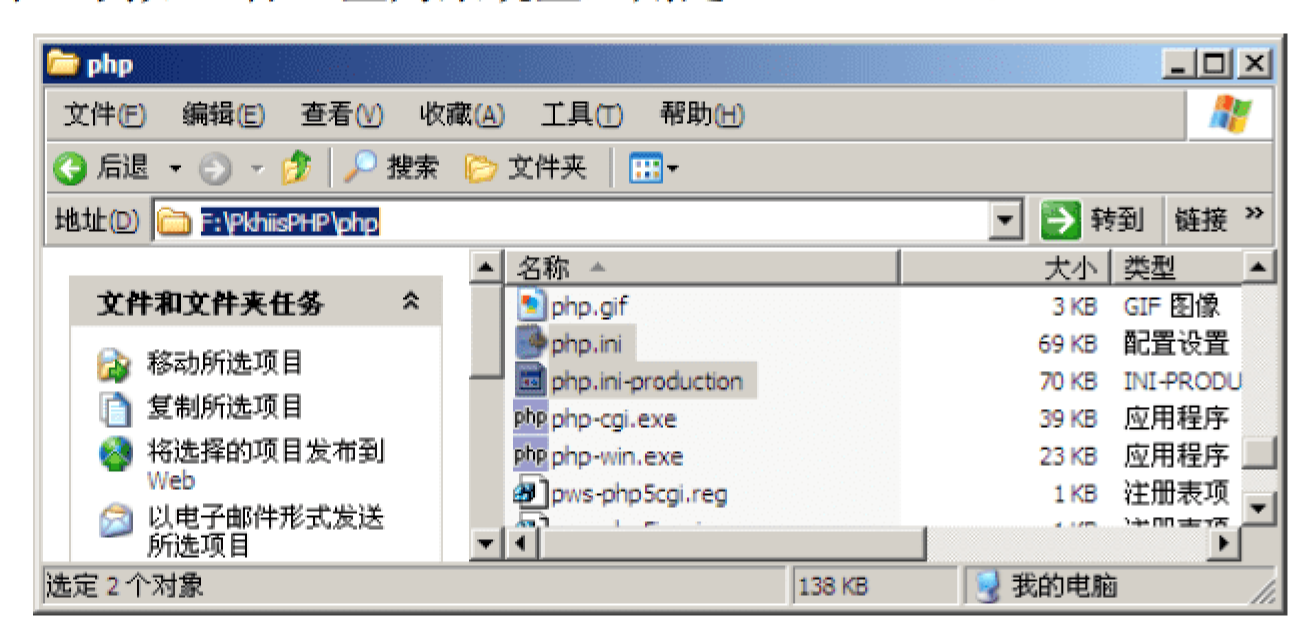


图 1.25　生成 php.ini

（4）打开 php.ini 文件并定位到 extension_dir = "./"这一行，修改为 extension_dir=" F:\PkhiisPHP\php\ext"。

（5）同样在 php.ini 文件中，将下面 5 项设置前的分号去掉，并设置相应的参数，完成对 CGI 和 FastCGI 的支持。

```
;cgi.force_redirect = 1
;cgi.fix_pathinfo=1
;fastcgi.impersonate = 1
;fastcgi.logging = 0
;cgi.rfc2616_headers = 1
```

（6）加载动态库。例如，定位到;extension=php_MySQL.dll 这一行，将前面的分号“;”去掉，完成 MySQL 动态库的加载。

（7）PHP 安装成功后，还要对 FastCGI 进行配置，在系统盘的\system32\inetsrv\文件夹下找到 fcgiext.ini 文件，在该文件的最后添加如下内容：

```
[Types]
php=PHP
[PHP]
ExePath=F:\PkhiisPHP\php\php-cgi.exe
```

其中，[Types]指定语言为 PHP；[PHP]指定 php-cgi.exe 在本机中的安装位置。

秘笈心法

心法领悟 008：使用 IIS 和 PHP 5.3.1 来配置 PHP 的开发环境。

如果使用 IIS 和 PHP 5.3.1 来配置 PHP 的开发环境，那么本机中必须安装微软的 VC++ 2008（Visual C++ 2008，简称 VC++ 2008）的运行库（vcredist_x64.exe 或者 vcredist_x86.exe）和 FastCGI 扩展（fcgisetup32.msi），以确保本机的 IIS 6 支持 FastCGI。

实例 009	安装 MySQL 光盘位置：光盘\MR\01\009	初级 趣味指数：★★★☆

实例说明

MySQL 是目前最为流行的开放源码的数据库，是完全网络化的跨平台的关系型数据库系统。它是由 MySQL AB 公司开发、发布并支持的。任何人都能从 Internet 上下载 MySQL 软件，并且不需支付任何费用。

在 MySQL 的官方网站（http://www.mysql.com）中可以下载到最新版本的 MySQL 服务器。

■ 关键技术

在 MySQL 的官方网站（http://www.mysql.com）中下载最新版本的 MySQL 服务器的具体步骤如下：

（1）打开 IE 浏览器，在地址栏中输入 http://www.mysql.com，进入 MySQL 的官方网站，如图 1.26 所示。



图 1.26　MySQL 的官方网站

（2）单击 Downloads 超链接，进入 MySQL 的 Downloads 页面，如图 1.27 所示。

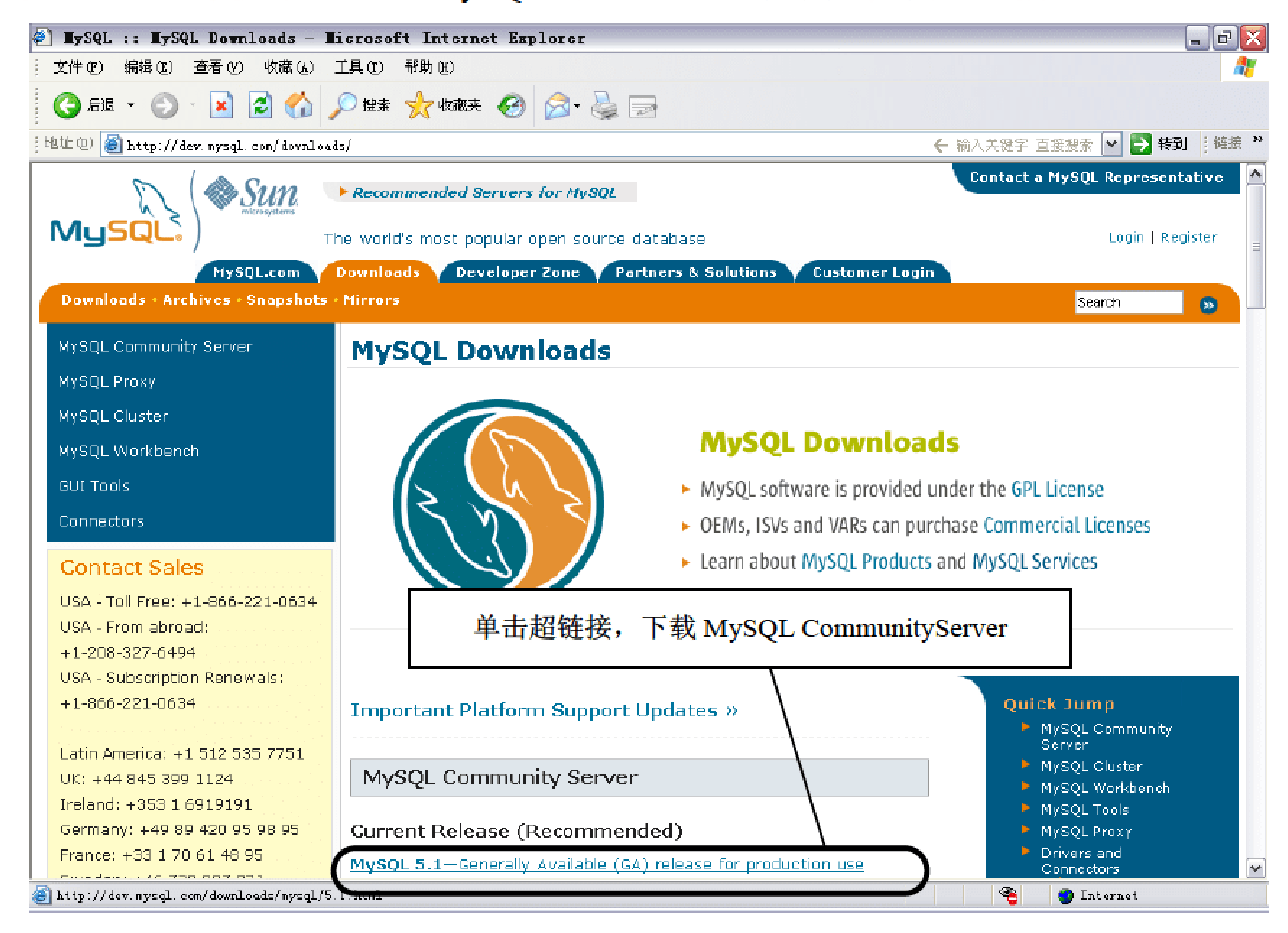
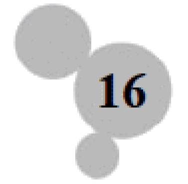


图 1.27　MySQL Downloads 页面



（3）单击 MySQL 5.1—Generally Available（GA）release for production use 超链接，进入 MySQL Connmunity Server 下载页面，如图 1.28 所示。

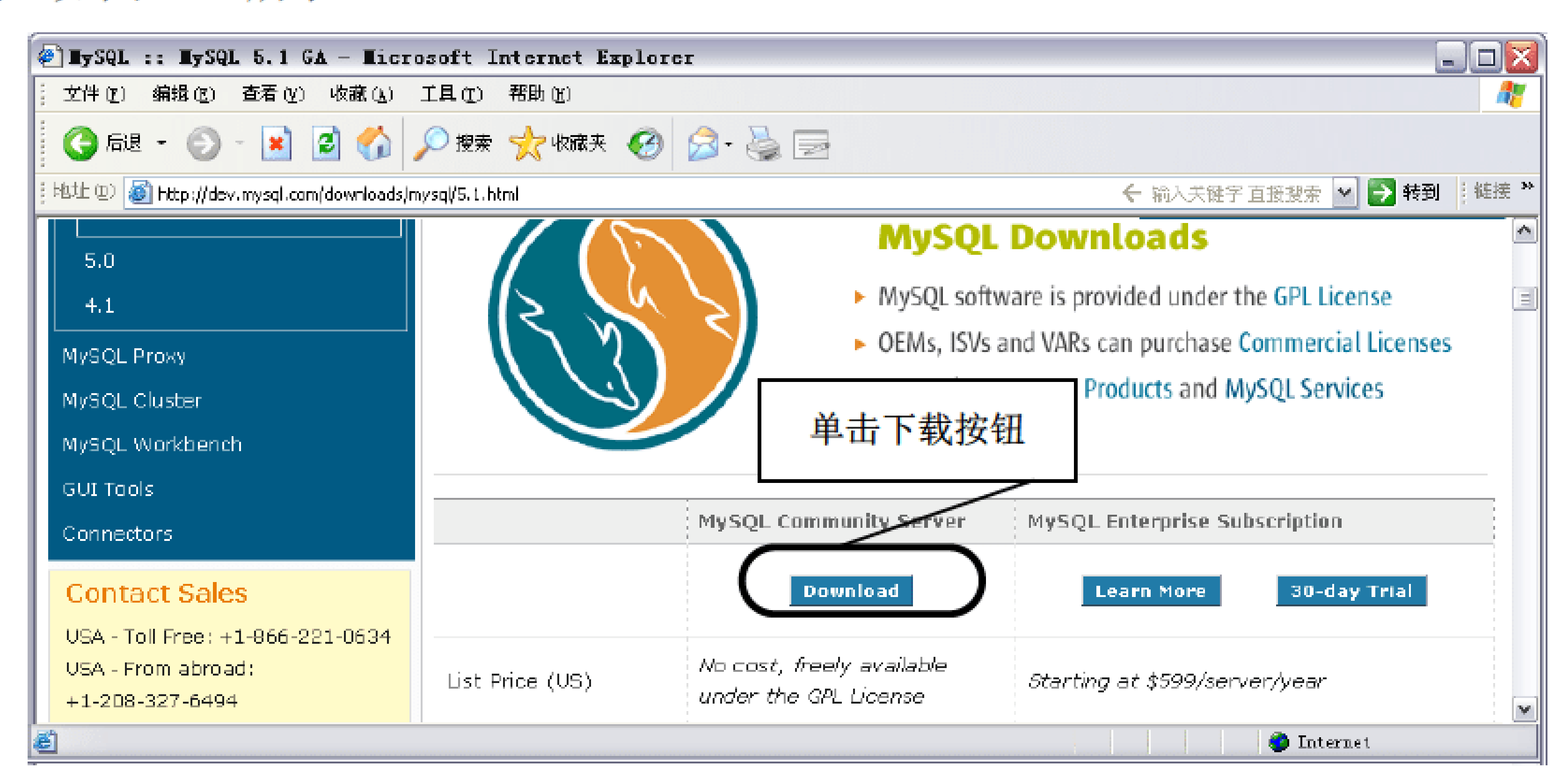


图 1.28　MySQL Connmunity Server 下载页面

（4）单击 Download 按钮，进入如图 1.29 所示页面。在该页面中，可以选择安装的平台。

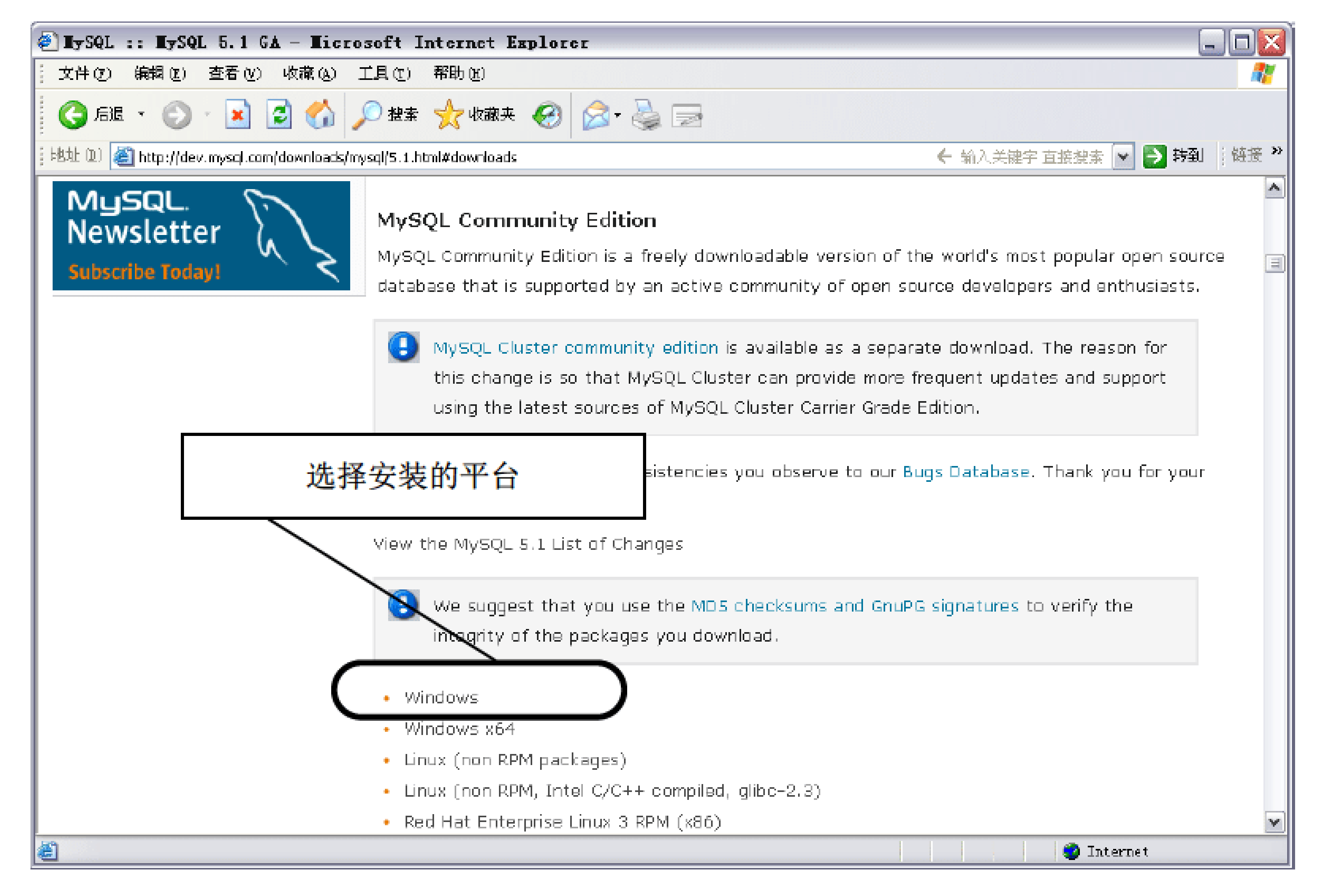


图 1.29　选择安装的平台

（5）这里选择 Windows 平台，单击 Windows 超链接，进入如图 1.30 所示的页面。在该页面中，可以下载不同安装方式的 MySQL。

（6）单击 Windows Essentials(x86)后面的 Download 超链接，将弹出下载对话框，在该对话框中单击“保存”按钮，下载 Windows 安装版的 MySQL。

说明：（1）如果在该页面中没有出现 Download 超链接，可以单击 Pick a mirror 超链接，选择一个镜像进行下载。

（2）本书在讲解 MySQL 的下载和安装时使用的是 mysql-essential-5.1.39 版本，当读者看到本书时，这个版本可能不是最新的版本了。不过 MySQL 的下载和安装步骤基本没有变化，读者仍然可以使

用本书中提供的步骤、方法进行下载和安装。

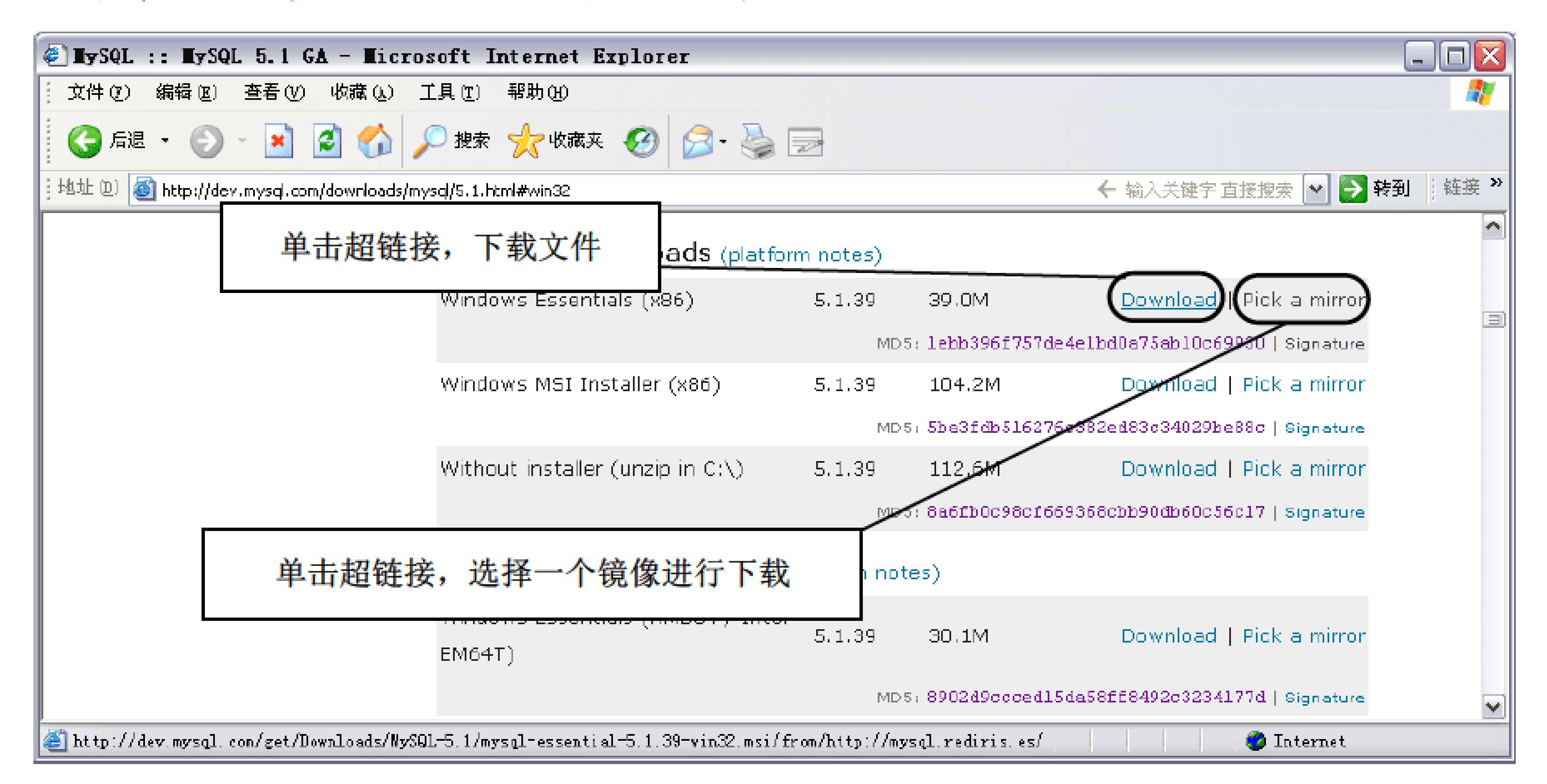


图 1.30　MySQL 的下载列表

设计过程

下面讲解 MySQL 的具体安装步骤。

（1）双击下载后的 mysql-essential-5.1.39-win32.msi 文件，打开安装向导对话框，单击 Next 按钮后，进入 Setup Type（选择安装类型）对话框。在该对话框中，共包括 Typical（典型）、Complete（完全）和 Custom（自定义）3 种安装类型。这里选择典型安装，如图 1.31 所示。

（2）单击 Next 按钮，打开准备安装对话框，在该对话框中将显示默认的安装路径，单击 Install 按钮，开始安装 MySQL。在安装的过程中，将显示 MySQL Enterprise 的宣传对话框，在该对话框中，单击 Next 按钮，进入下一个宣传对话框，再次单击 Next 按钮，将进入如图 1.32 所示的 Wizard Completed（成功向导）对话框。在该对话框中采用默认设置，单击 Finish 按钮，继续安装 MySQL 服务器。

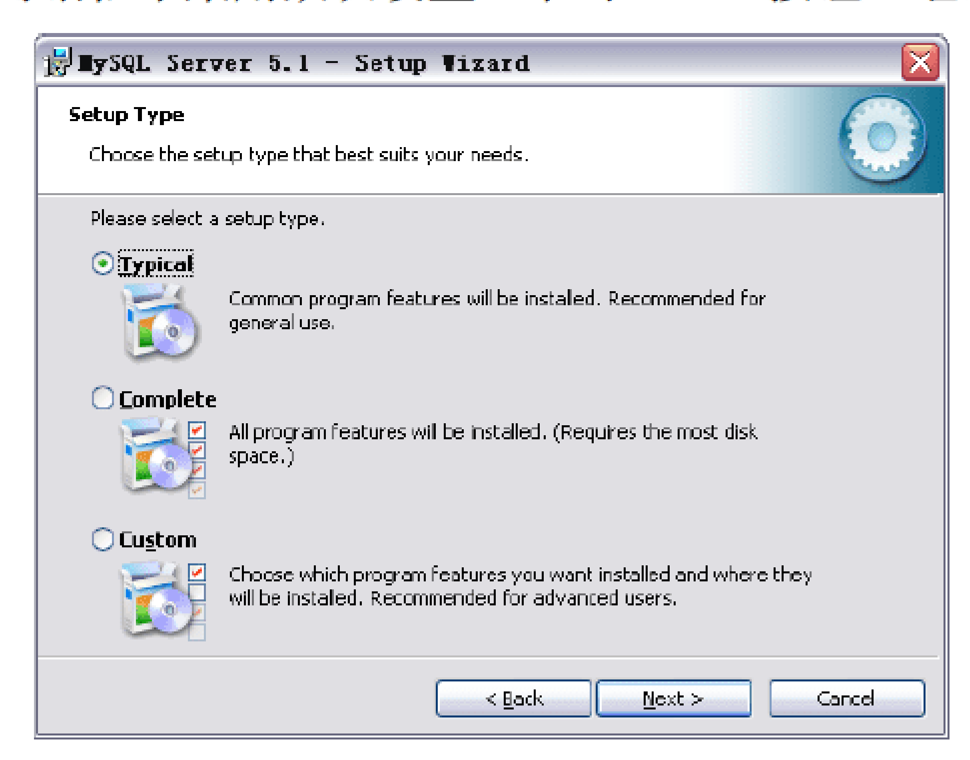



图 1.31　选择安装类型对话框

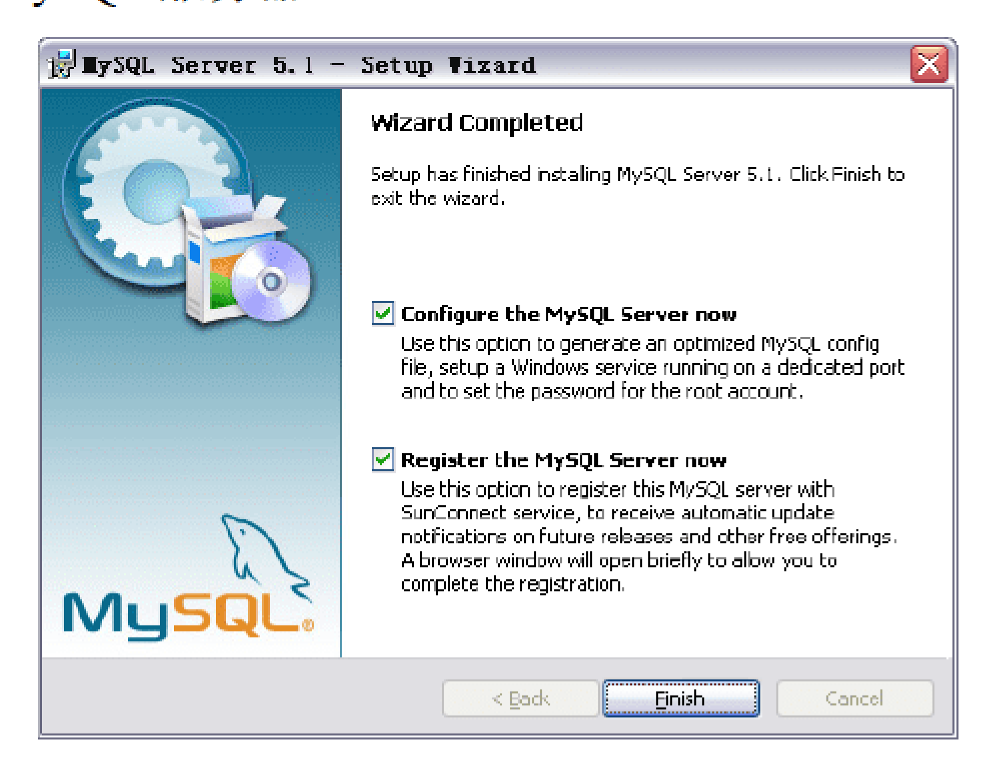
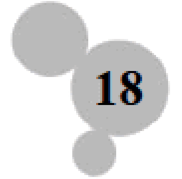


图 1.32　成功向导对话框

（3）这时将进入欢迎安装 MySQL 服务器的对话框，在该对话框中，单击 Next 按钮，进入如图 1.33 所示的 MySQL 服务器配置类型对话框。在该对话框中可以选择详细配置或者标准配置，这里选择详细配置。

（4）单击 Next 按钮，进入如图 1.34 所示的选择服务器类型对话框。在该对话框中将提供开发者类型、服务器类型和致力于 MySQL 服务类型。这里选择默认的开发者类型。

（5）单击 Next 按钮，进入如图 1.35 所示的选择数据库用法对话框，在该对话框中提供 3 个选项，分别是



多功能型（Multifunctional Database）、仅提供事务型（Transactional Database Only）和仅不提供事务型（Non-Transactional Database Only）。这里选择多功能型。

（6）单击 Next 按钮，进入如图 1.36 所示的设置服务器安装路径的对话框，这里采用默认设置。

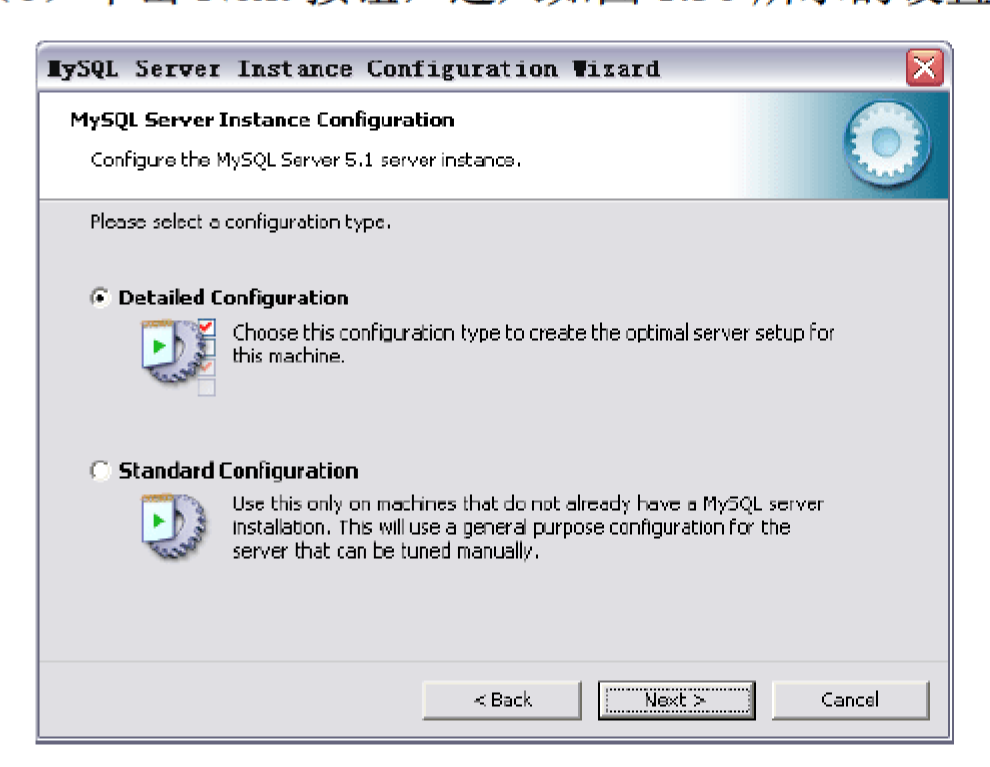


图 1.33　MySQL 服务器配置类型对话框

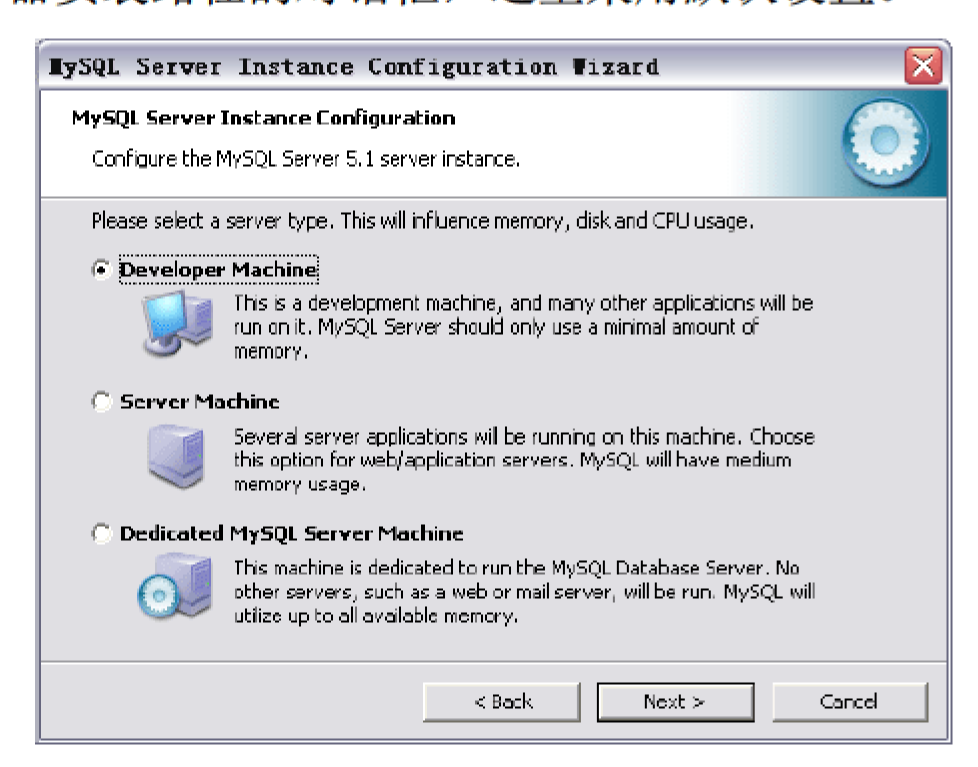


图 1.34　选择服务器类型对话框

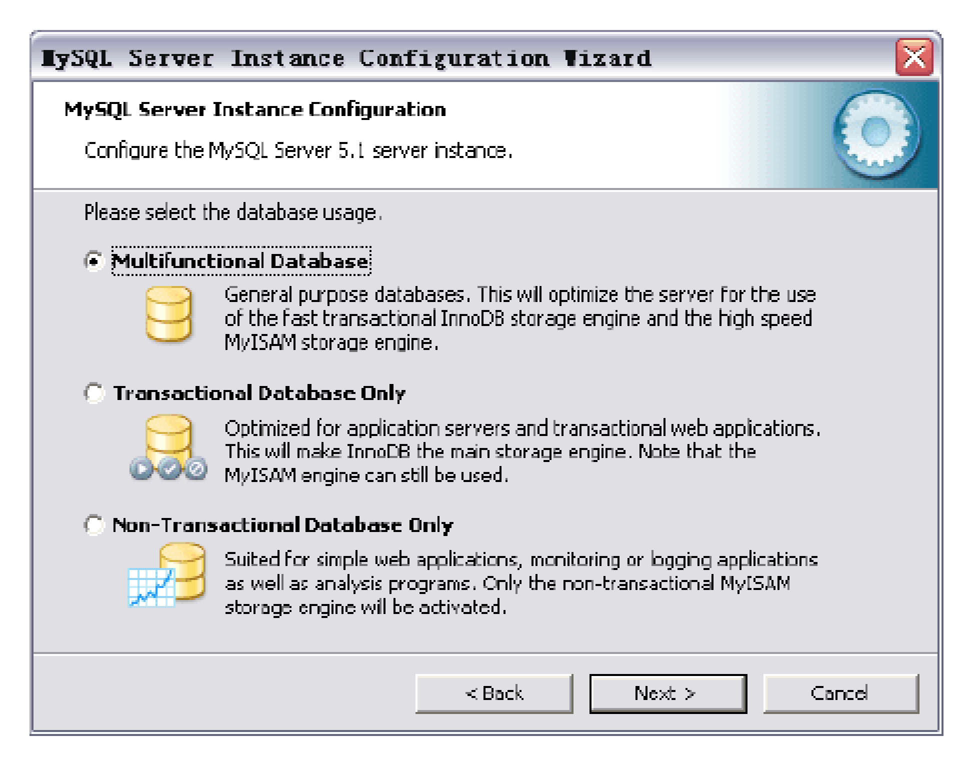


图 1.35　选择数据库用法对话框

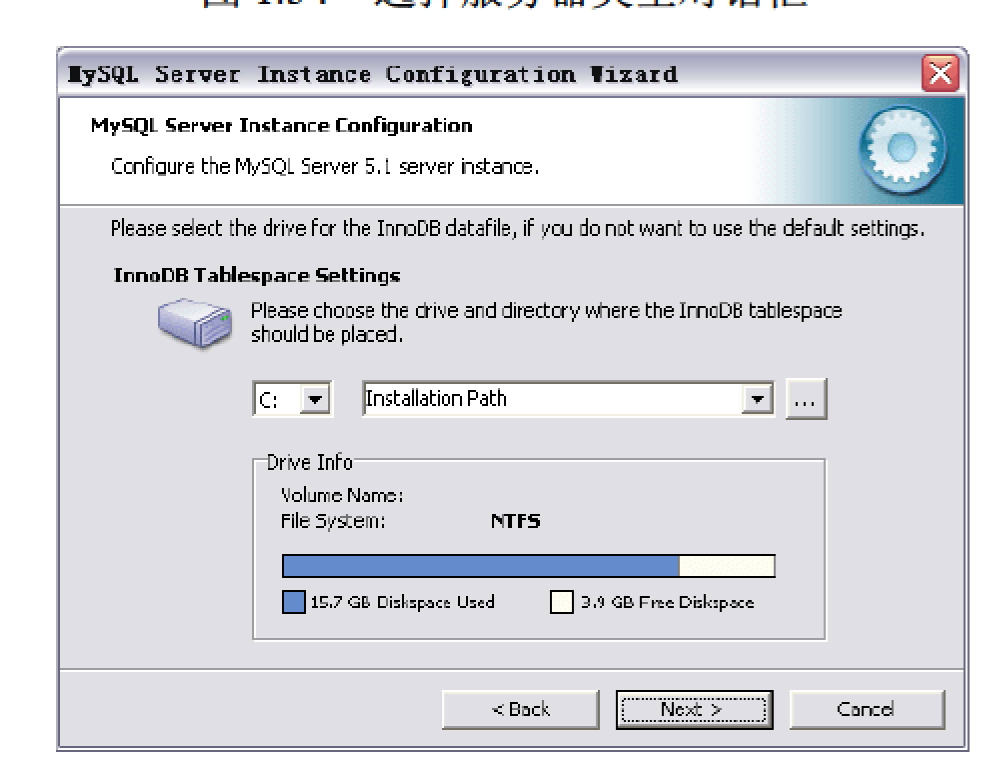


图 1.36　设置服务器安装路径对话框

（7）单击 Next 按钮，进入如图 1.37 所示的设置提示连接的数量的对话框，这里采用默认设置。

（8）单击 Next 按钮，进入如图 1.38 所示的设置网络选项对话框，在该对话框中采用默认设置。

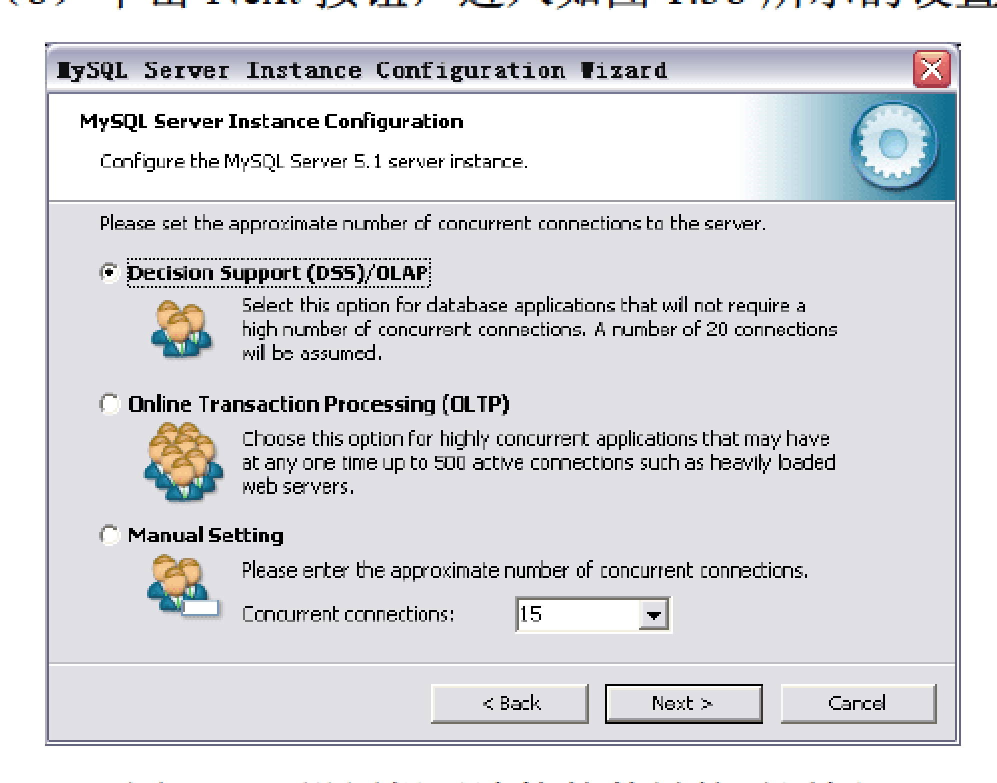


图 1.37　设置提示连接的数量的对话框

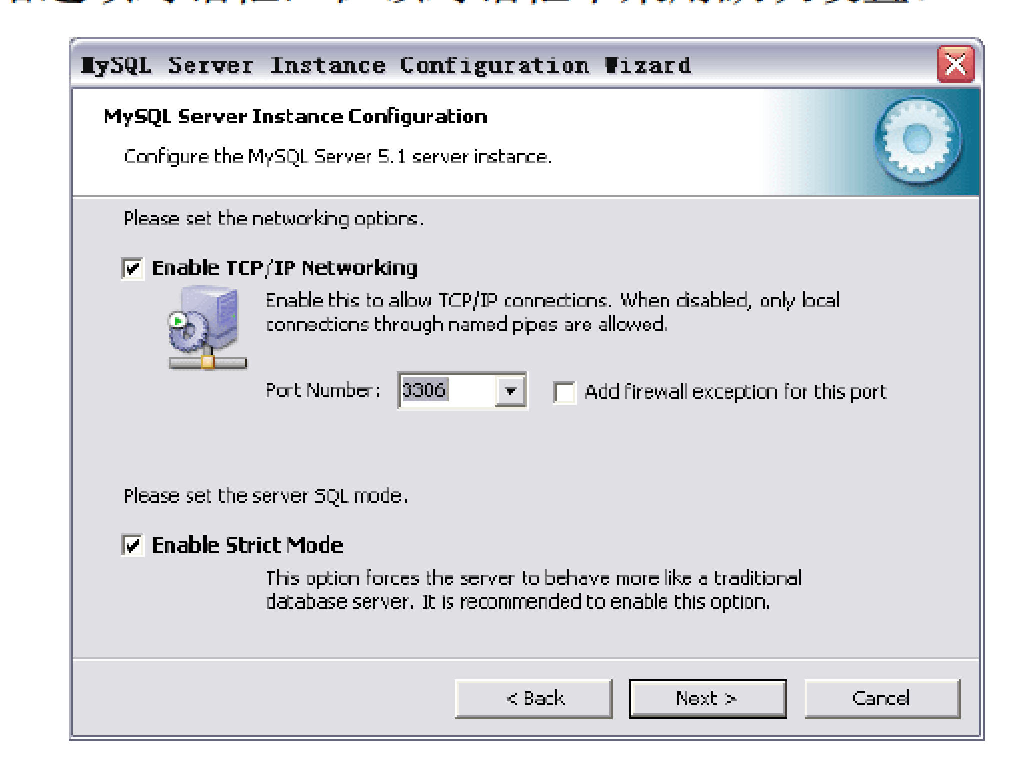


图 1.38　设置网络选项对话框

✍ **技巧：** MySQL 使用的默认端口是 3306，在安装时，可以修改为其他的，如 3307。但是一般情况下，不要修改默认的端口号，除非 3306 端口已经被占用。

（9）单击 Next 按钮，将打开设置默认的字符集的对话框，在该对话框中选中 Manual Selected Default Character Set/Collation 单选按钮，并在 Character Set（字符集）下拉列表框中选择 gbk（简体中文），如图 1.39 所示。

（10）单击 Next 按钮，进入如图 1.40 所示的设置 Windows 选项的对话框，这里采用默认设置。



图 1.39　设置默认的字符集对话框

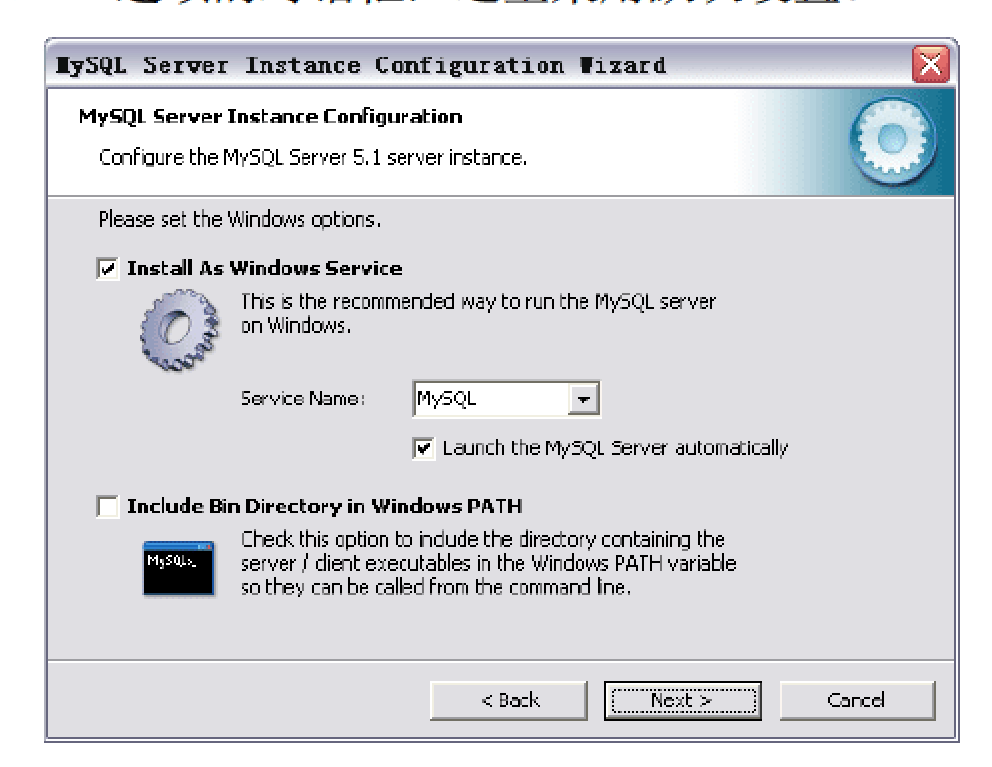


图 1.40　设置 Windows 选项对话框

（11）单击 Next 按钮，进入安全设置选项对话框，该对话框用于为默认用户 root 设置密码，这里设置为 111，如图 1.41 所示。

（12）单击 Next 按钮，进入准备执行对话框，单击 Execute 按钮，完成服务器配置，配置完成后，将显示如图 1.42 所示的对话框。单击 Finish 按钮，完成 MySQL 的安装。



图 1.41　安全设置选项对话框

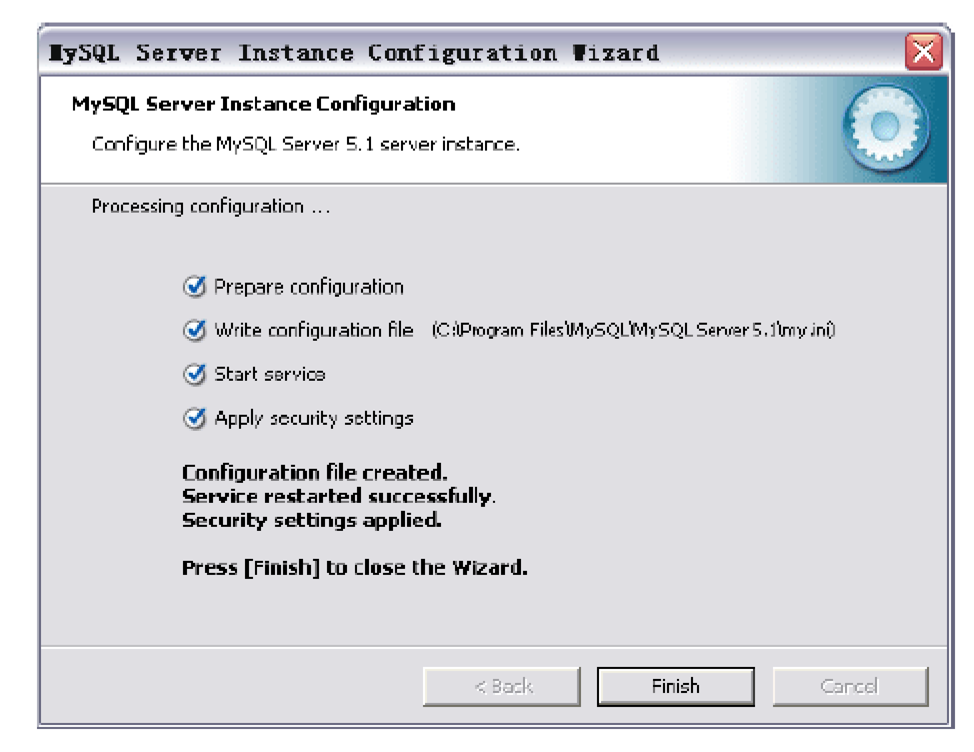
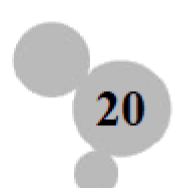


图 1.42　完成服务器配置

说明： MySQL 安装完成后，将在“开始”/“所有程序”菜单下添加一个 MySQL 项，在该项中包括一个 MySQL Server 5.1 子项，在该子项下，又包括 MySQL Command Line Client、MySQL Server Instance Config Wizard 和 SunInventory Registration 3 个子项。其中 MySQL Command Line Client 为 MySQL 的客户端命令行，通过该项，可以查看数据表结构及执行 SQL 语句；MySQL Server Instance Config Wizard 项可用于修改 MySQL 的配置，如修改字符集和 root 用户的密码等。

至此，MySQL 安装成功，如果要查看 MySQL 的安装配置信息，可以通过 MySQL 安装目录下的 my.ini 文件来完成。

在 my.ini 文件中，可以查看到 MySQL 服务器的端口号、MySQL 在本机的安装位置、MySQL 数据库文件存储的位置，以及 MySQL 数据库的编码等配置信息，如图 1.43 所示。



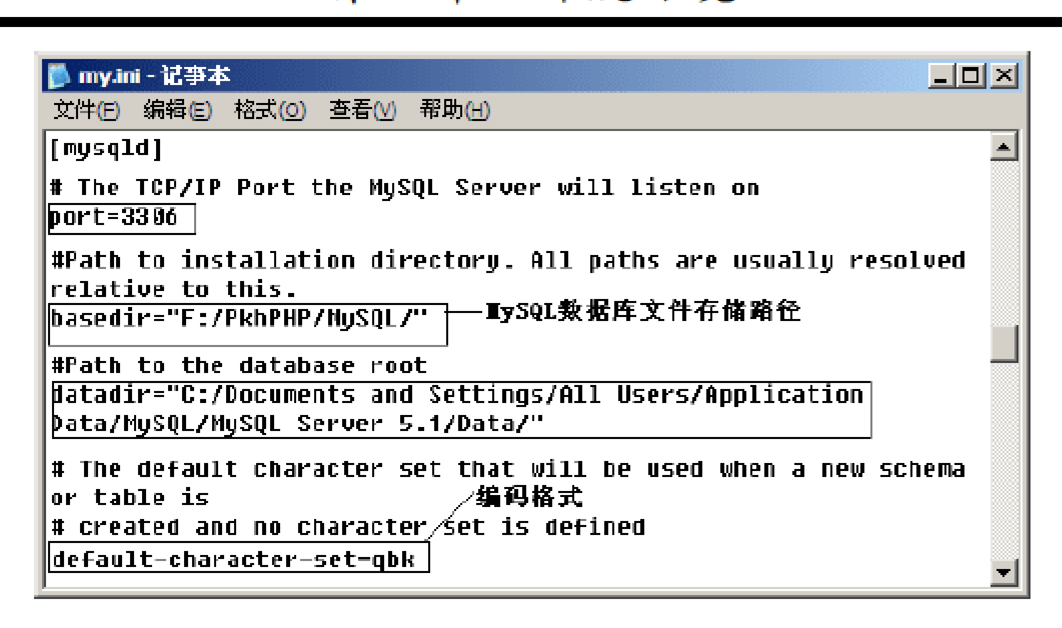


图 1.43　my.ini 文件的配置信息

秘笈心法

心法领悟 009：设置 MySQL 用户的密码。

在安装 MySQL 数据库时，一定要牢记在第（1）步中设置的 root 用户的密码，它是我们在访问 MySQL 数据库时必须使用的。



实例说明

本书以 Windows 2003 中使用的 IIS 6.0 为例，讲解 IIS 的安装。至于在其他操作系统中的安装步骤是相同的，只是使用的版本有所不同。

关键技术

IIS 的安装非常简单，关键是如何配置 IIS 服务器。

（1）选择“开始”/“设置”/“控制面板”命令，在弹出的窗口中双击“管理工具”图标，在弹出的窗口中单击“Internet 信息服务（IIS）管理器”图标，将弹出 IIS 管理器界面，如图 1.44 所示。

（2）图中的“默认网站”即为 IIS 服务器的文件目录。右击“默认网站”，在弹出的快捷菜单中选择“属性”命令，弹出“默认网站 属性”对话框，可在其中设置网站的属性，如图 1.45 所示。

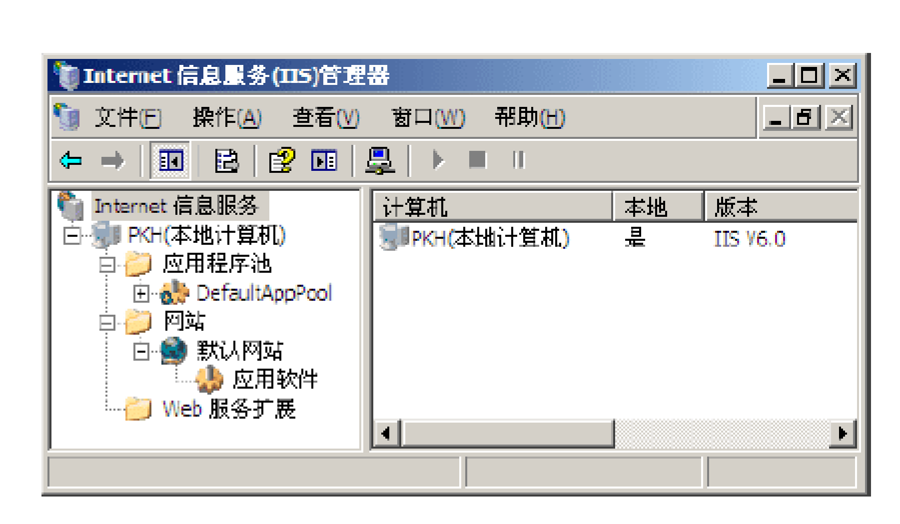


图 1.44　IIS 管理器界面



图 1.45　“默认网站 属性”对话框

（3）默认选中的是“网站”选项卡，在其中可以设置 TCP 端口号，这里将 IIS 端口号设置为“82”。设置完成后选择“主目录”选项卡，如图 1.46 所示。

（4）在“主目录”选项卡中进行 3 项设置：第 1 项是设置本地路径，用户可以自行定义，如 F:\PkhiisPHP\www；第 2 项是设置执行权限，这里设置为纯脚本；第 3 项是添加 PHP 的支持。单击“主目录”选项卡中的“配置”按钮，弹出如图 1.47 所示的“应用程序配置”对话框。



图 1.46　“主目录”选项卡

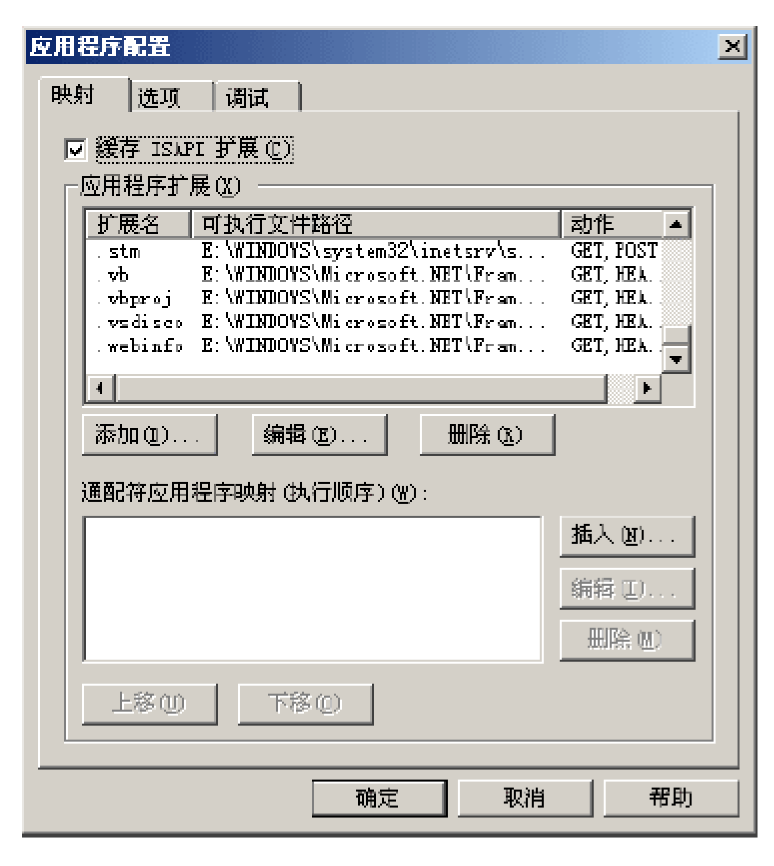


图 1.47　应用程序配置

（5）选择“映射”选项卡，完成添加、编辑应用程序扩展。单击“添加”按钮，在弹出的如图 1.48 所示对话框中首先指定可执行文件，这里指定的是存储在系统盘的 system32\inetsrv\文件夹下的 fcgiext.dll 文件，然后设置扩展名为“.php”，最后单击“确定”按钮，返回到“默认网站 属性”对话框。

（6）选择“文档”选项卡，添加一个.php 的默认页。添加后的效果如图 1.49 所示。



图 1.48　添加 PHP 扩展

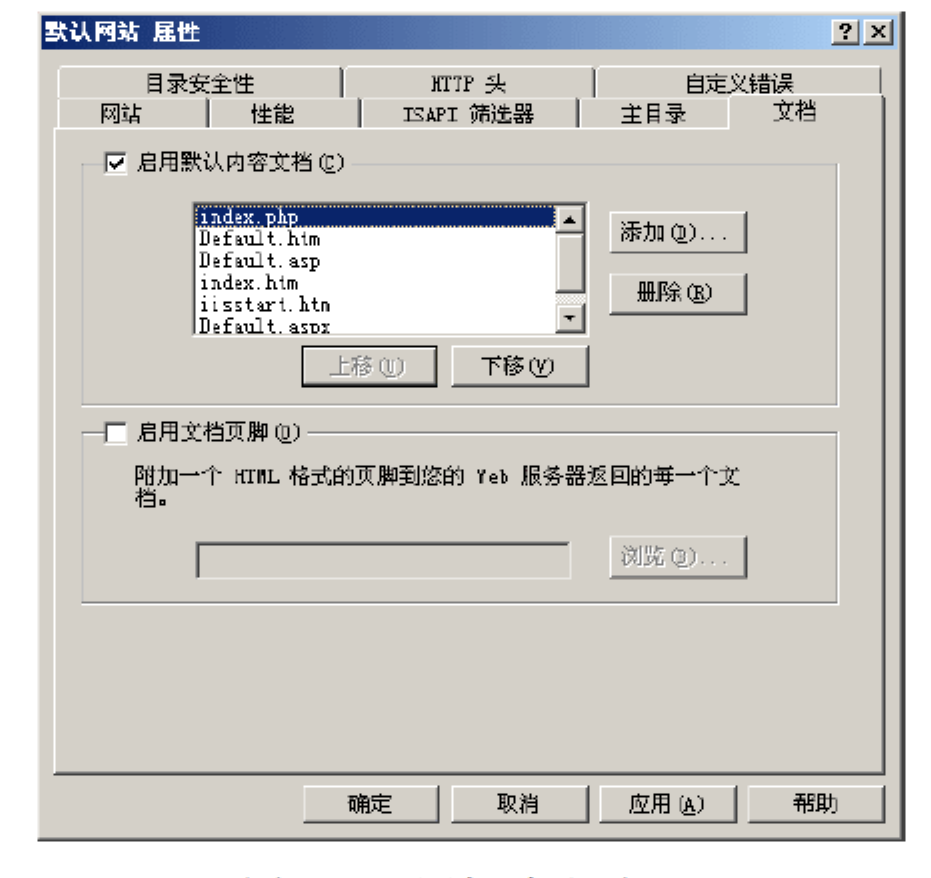


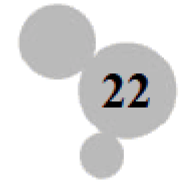
图 1.49　添加默认页

（7）关闭所有对话框，重启 IIS 服务器。

设计过程

（1）选择“开始”/“设置”/“控制面板”命令，在弹出的窗口中单击“添加/删除程序”按钮。

（2）单击左侧的“添加/删除 Windows 组件”按钮，弹出“Windows 组件向导”对话框。依次选中并双击“Windows 组件向导”对话框中的“应用程序服务器”选项、“应用程序服务器”窗口中的“Internet 信息服务



（IIS）”选项、“Internet 信息服务（IIS）”对话框中的“万维网服务”选项。最后选中“万维网服务”复选框。

（3）设置完毕后单击“确定”按钮。具体操作过程如图 1.50 所示。

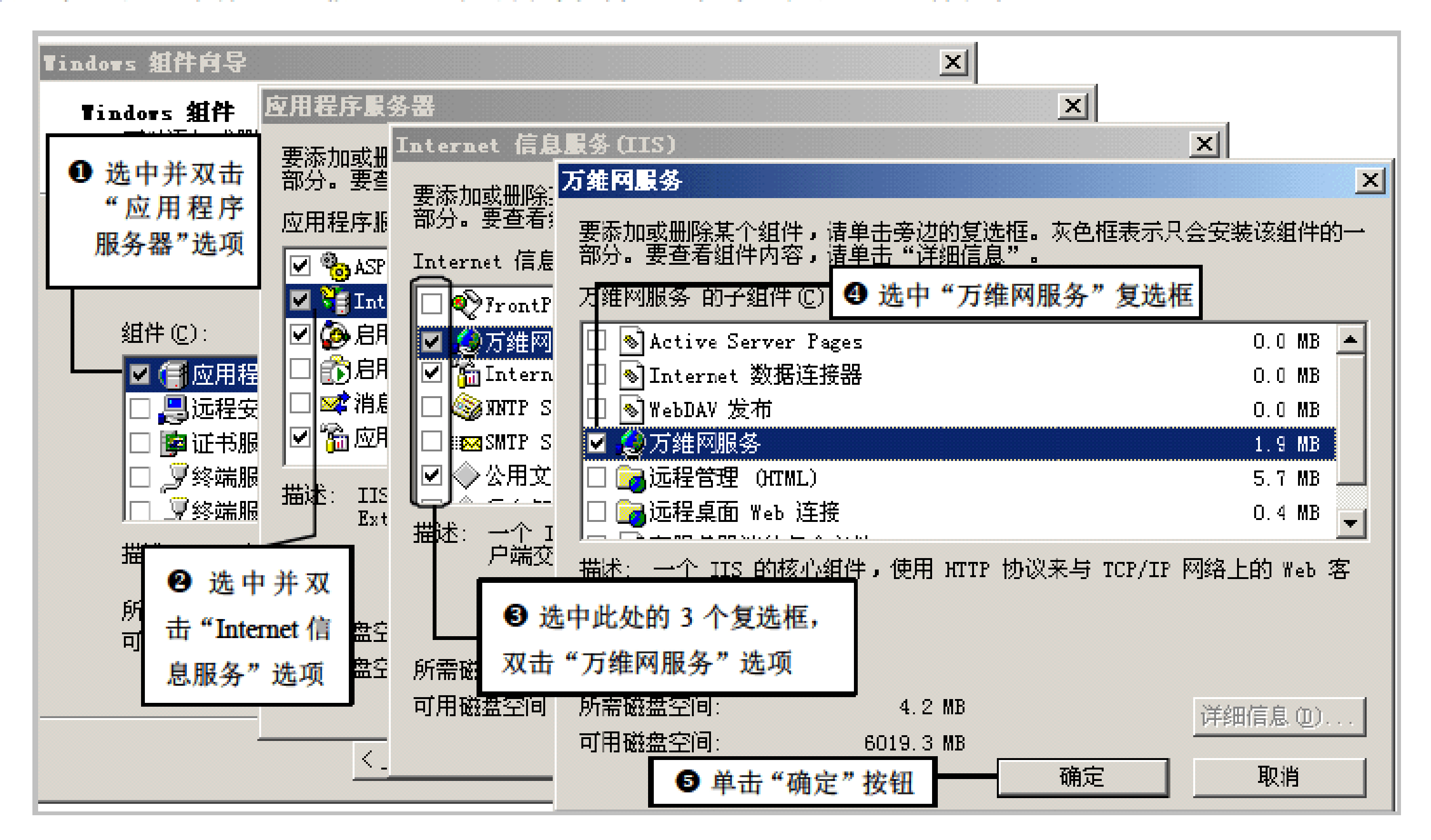


图 1.50　Internet 信息服务（IIS）

安装完成后即可使用 IIS。

秘笈心法

心法领悟 010：测试 IIS 是否安装成功。

安装 IIS 后，即可在 IE 浏览器的地址栏中输入 http://localhost，测试 IIS 是否安装成功。

如果本机的 80 端口被 Apache 占用，那么就需要修改 IIS 的端口号。例如修改为 82 端口，那么测试时应该在地址栏中输入 http://localhost:82/。



实例说明

本实例编写第 3 个 PHP 脚本，在 IIS 服务器下运行，测试通过 IIS+PHP+MySQL 搭建的 PHP 开发环境是否成功，运行结果如图 1.51 所示。

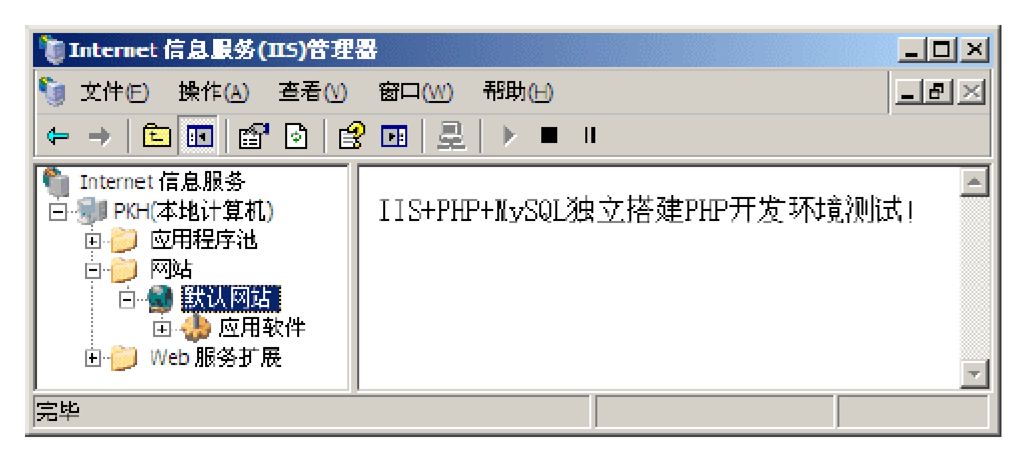


图 1.51　测试独立环境是否搭建成功

关键技术

确定 IIS 服务器设置的本地路径，这是在 IIS 的配置中完成的，必须将 PHP 脚本存储于 IIS 设置的本地路径下才能够运行。

设计过程

（1）在 IIS 指定的本地路径下（F:\PkhiisPHP\www）新建一个文本文档，在该文档中输入如下内容：

```
<?php
echo "IIS+PHP+MySQL 独立搭建 PHP 开发环境测试！";
?>
```

（2）将该文档重命名为 index.php。

（3）选择“开始”/“设置”/“控制面板”命令，在弹出的窗口中双击“管理工具”图标，然后选择“Internet 信息服务（IIS）管理器”选项，依次展开“本地计算机”和“网站”节点，右击“默认网站”选项，在弹出的快捷菜单中选择“浏览”命令，若在“Internet 信息服务（IIS）管理器”窗口的右侧输出如图 1.51 所示的内容，说明独立环境搭建成功。

秘笈心法

心法领悟 011：IIS 中的 UrlRewrite 技术。

UrlRewrite 即地址变换，可以实现静态功能，方便搜索引擎收录，比如 http://***/news.php?id=1 可以变为 htt://***/news-1.html。不但可以实现开发标记的隐藏，而且避免每个页面都输出.php 后缀，进而提高网站的安全性。

IIS 默认是不支持此功能的，需要单独安装插件 ISAPI_Rewrite3_0048_Lite.msi。安装完成后打开 IIS，在 Web 属性的“ISAPI 筛选器”里添加位于安装目录的 Helicon\ISAPI_Rewrite3 文件夹里的 ISAPI_Rewrite.dll 即可，名称可以自定义。

1.4　LAMP——独立配置 PHP 开发环境

在 Linux 下搭建 PHP 环境比在 Windows 中要复杂一些，除了 Apache、PHP 等软件外，还要安装一些相关工具，并设置必要参数。这里给出在 Linux 下搭建 PHP 环境的必要步骤，如果用户在安装过程中遇到特殊的问题，还需要翻阅与 Linux 相关的资料、书籍或者手册。

安装之前要准备的安装包如下：

- ☑ apache_1.3.41.tar.gz
- ☑ php-5.0.0.tar.gz
- ☑ mysql-standard-5.0.0-alpha-pc-linux-i686.tar.gz
- ☑ libxml2-2.6.19.tar.gz

说明：本书中在讲解 Linux 下搭建 PHP 开发环境时，使用的软件版本相对较低，如果读者想要下载这些软件，可以登录 http://221.8.65.77/mrbook/bookzy/linux.rar。

这里首先在 Linux 系统的/usr/local/文件夹下新建一个 work 文件夹，并将使用的软件全部存储在此文件夹下。

实例 012	安装 Apache 服务器 光盘位置：光盘\MR\01\012	中级 趣味指数：★★★☆

实例说明

本实例讲解在 Linux 操作系统下安装 Apache 服务器。Apache 服务器安装成功后测试的运行结果如图 1.52 所示。



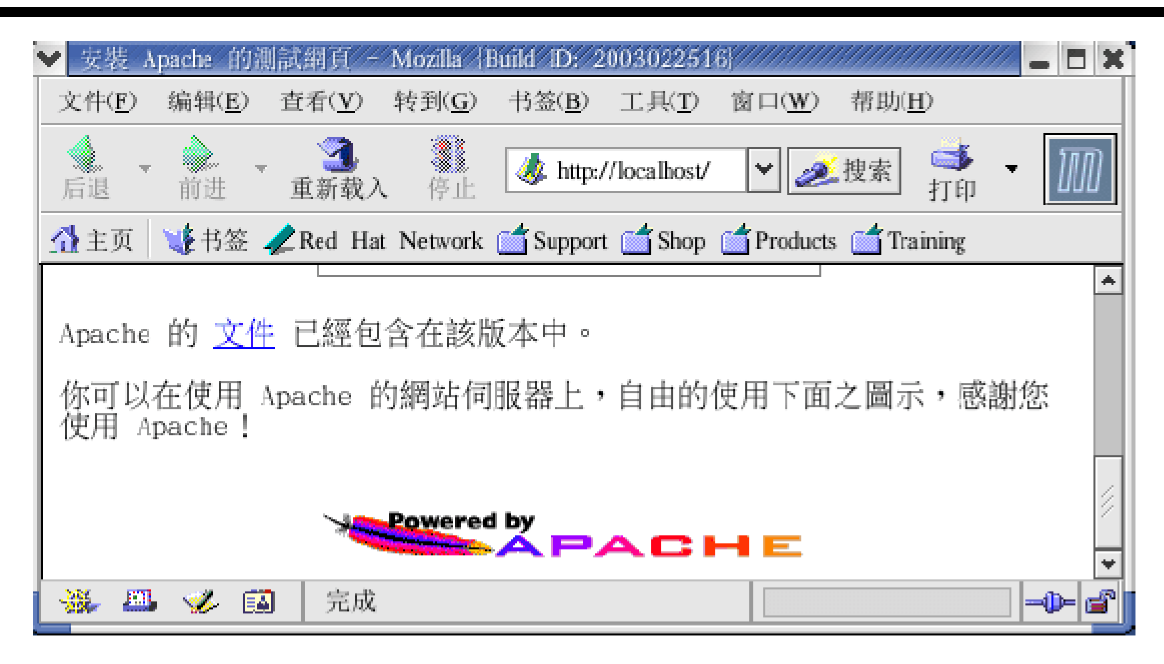


图 1.52　Apache 配置成功

关键技术

Linux 系统下配置 Apache 服务器使用的仍是 httpd.conf 文件，该文件位于/usr/local/apache/conf 中。

（1）在 httpd.conf 文件中，定位到如图 1.53 所示的位置，添加相应的内容，并设置 PHP 文件的后缀。

（2）同样在 httpd.conf 文件中，定位到如图 1.54 所示的位置，设置 PHP 文件的默认页。



图 1.53　设置文件后缀



图 1.54　设置默认文件

设计过程

安装 Apache 服务器，首先需要打开 Linux 终端（Linux 下几乎所有的软件都需要在终端下安装）。选择 RedHat9 的“主菜单”/“系统工具”命令，在弹出的子菜单中选择“终端”命令。Apache 服务器的安装步骤如下：

（1）解压 apache_1.3.41.tar.gz 压缩包，将其存储到/url/local/apache 目录下，执行的命令如图 1.55 所示。

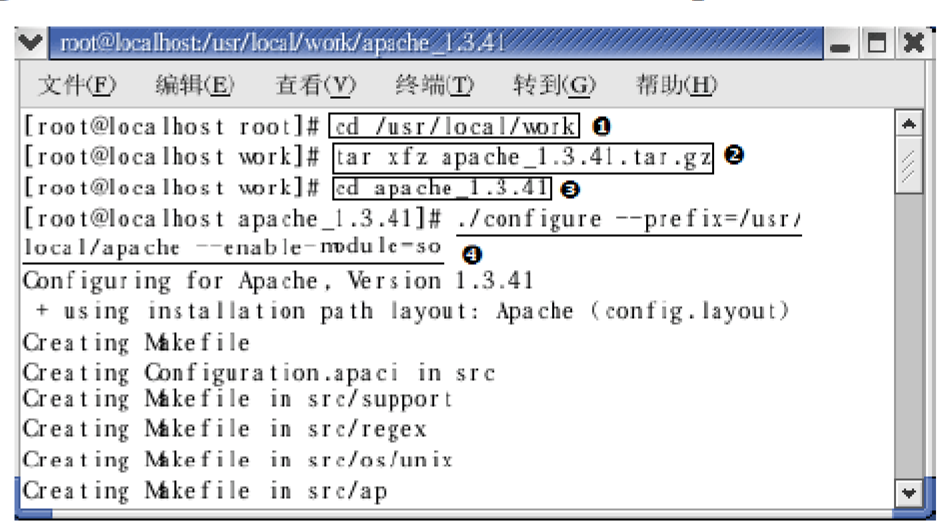
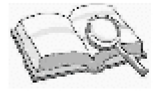


图 1.55　输入命令



代码导读

❶ 进入 Apache 安装文件的目录。

❷ 解压 Apache 服务器的安装包。

❸ 进入 apache_1.3.41 目录。

❹ 将 Apache 服务器安装到/usr/local/apache 目录下。

（2）执行 make 命令，编译 apache 文件。执行命令如图 1.56 所示。

（3）执行 make install 命令，开始安装 Apache 服务器。执行命令如图 1.57 所示。

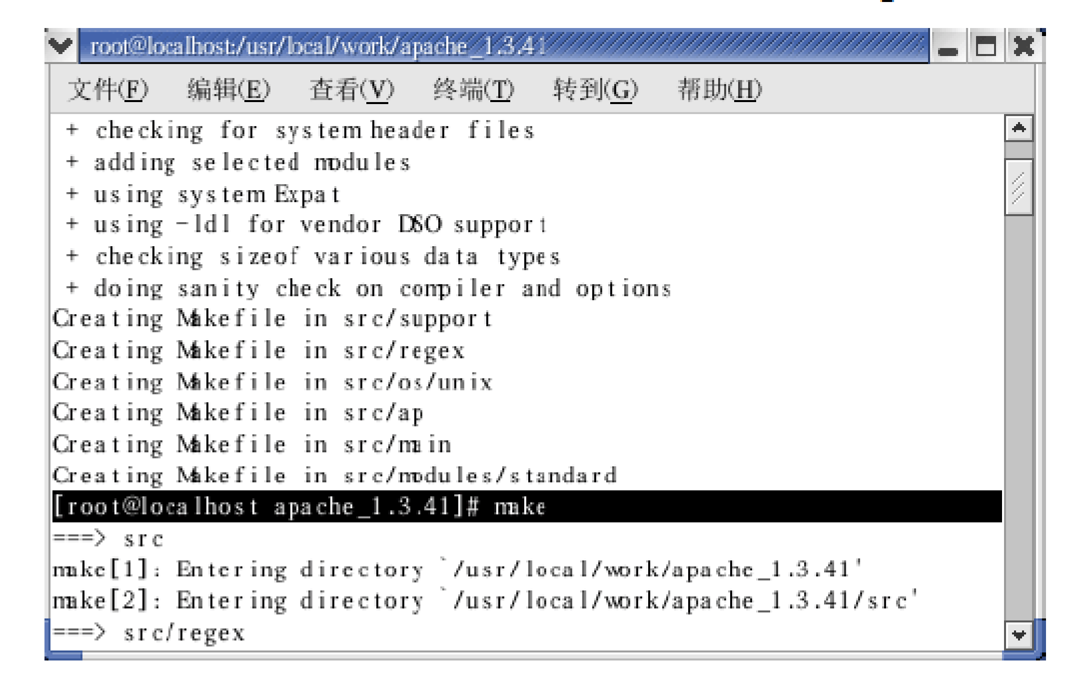


图 1.56　执行 make 命令编译文件

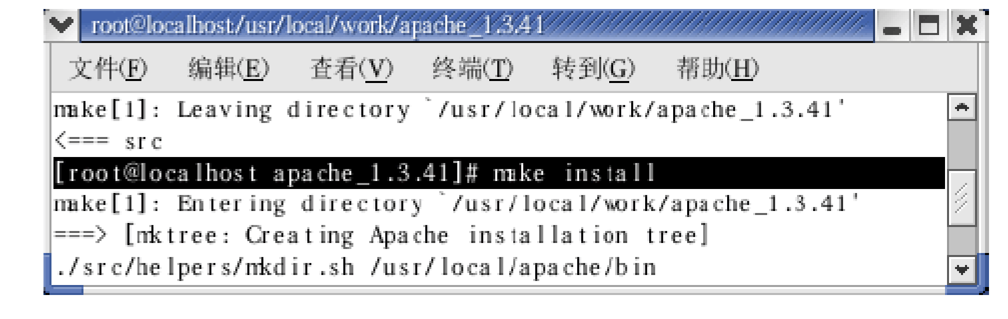


图 1.57　执行 make install 安装命令

（4）Apache 服务器安装成功后，将给出如图 1.58 所示的命令提示。



图 1.58　安装成功返回命令

秘笈心法

心法领悟 012：Linux 下启动 Apache。

安装成功后，就可以启动 Apache 服务器了，启动 Apache 服务器使用的命令如图 1.59 所示。



图 1.59　执行启动 Apache 服务器的命令

实例 013　安装 MySQL 数据库服务器

光盘位置：光盘\MR\01\013

中级

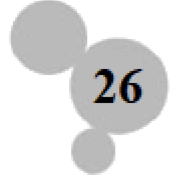
趣味指数：★★★★☆

实例说明

本实例讲解在 Linux 下安装 MySQL 数据库，使用的版本是 mysql-standard-5.0.0- alpha- pc- linux-i686.tar.gz。

关键技术

（1）在 Linux 操作系统中，备份和还原 MySQL 使用的命令如下：



备份：

```
mysqldump –uroot –proot –R  数据库名称>C:/data.sql
```

还原：

```
mysql –uroot –proot –D  数据库名称< C:/data.sql
```

（2）在 Linux 操作系统中，设置和修改 root 用户密码的命令如下：

设置密码：

```
mysqladmin –uroot password root
```

修改密码：

```
mysqladmin –uroot –p  旧密码  password  新密码
```

设计过程

下面讲解其具体的安装步骤。

（1）安装 MySQL 数据库同样要在 Linux 终端的命令中进行。首先创建 MySQL 账号，并将新建账号加入到组群，具体使用的操作命令如图 1.60 所示。

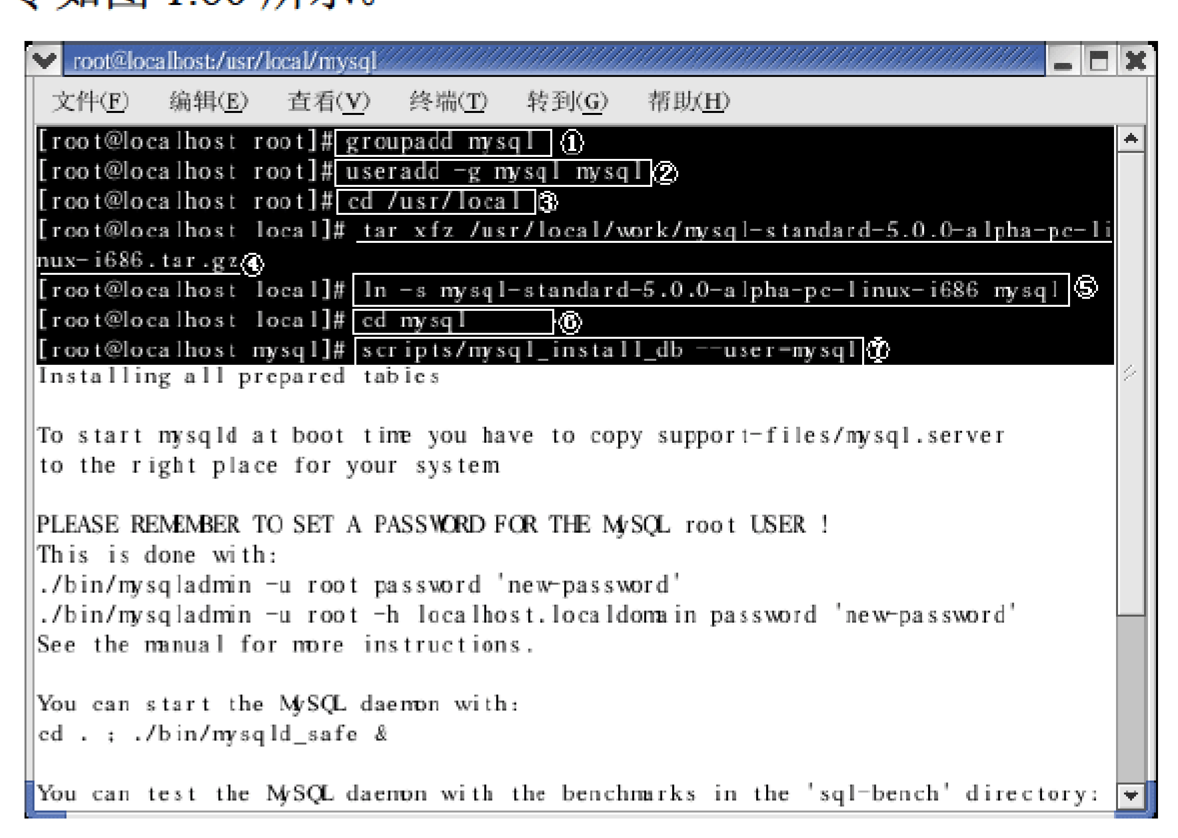
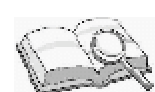


图 1.60　MySQL 的安装命令



代码导读

❶ 创建 MySQL 账号。

❷ 加入组群。

❸ 进入 MySQL 安装文件的目录。

❹ 解压 MySQL 服务器软件。

❺ 以链接的方式建立/usr/local/mysql 目录。

❻ 进入 MySQL 目录。

❼ 在/usr/local/mysql/data 中建立 MySQL 数据库。

（2）在完成数据库的安装之后，修改文件的权限。具体操作命令如图 1.61 所示。

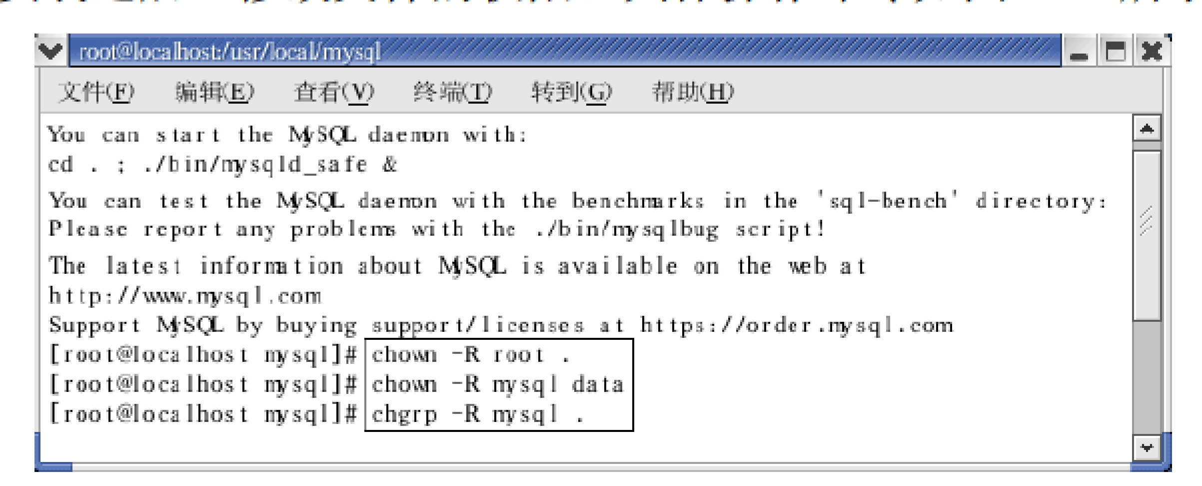


图 1.61　修改文件权限的命令

（3）完成权限修改后，看一下启动 MySQL 服务器的命令。具体命令如图 1.62 所示。



图 1.62　启动 MySQL 的命令

秘笈心法

心法领悟 013：Linux 下修改 MySQL 数据库用户的密码。

可以通过命令修改 MySQL 数据库用户的密码。在如图 1.63 所示的命令中，首先以无密码的方式登录 MySQL 数据库，然后通过命令设置 MySQL 数据库用户的密码，最后通过新设置的密码重新登录 MySQL 数据库。



图 1.63　登录 MySQL 和设置 MySQL 登录密码

实例 014　安装 PHP 5.0　光盘位置：光盘\MR\01\014　中级　趣味指数：★★★☆

实例说明

本实例讲解在 Linux 操作系统下安装 PHP 5.0。安装之前，首先需要查看 libxml 的版本号，如果小于 2.5.10，则需要先安装 libxml 的高版本，否则，可以直接安装 PHP 5.0。

关键技术

这里以 libxml2-2.6.19.tar.gz 为例，讲解 libxml 的安装步骤。

（1）将 libxml2-2.6.19.tar.gz 进行解压，并将其安装到指定的文件夹下，其操作命令如图 1.64 所示。

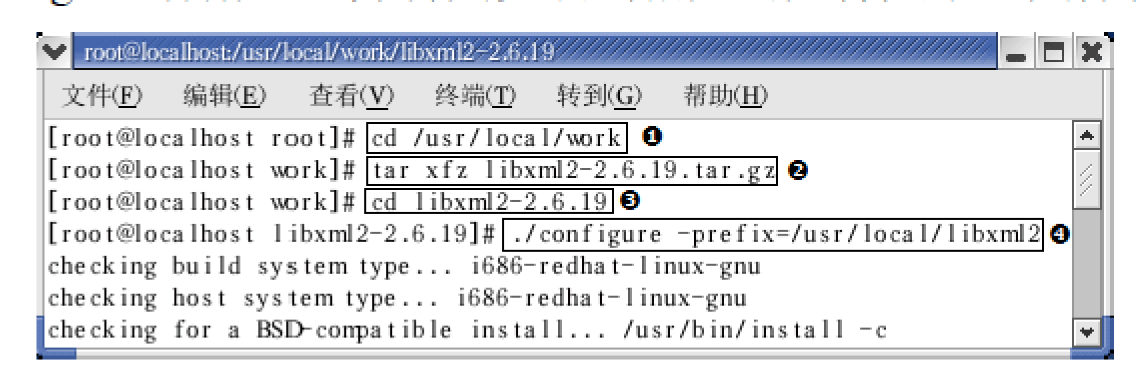
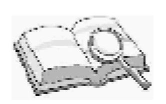
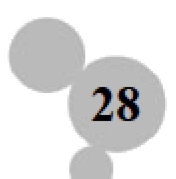


图 1.64　解压 libxml



代码导读

❶ 进入/usr/local/work 文件夹（libxml2-2.6.19 压缩包存储的位置）。



❷ 解压 libxml2-2.6.19.tar.gz。
❸ 进入 libxml2-2.6.19 文件夹。
❹ 将 libxml2-2.6.19 安装到/usr/local/libxml2。

（2）完成 libxml 的解压后，开始编译文件，使用的命令如图 1.65 所示。
（3）文件编译完成后，就可以执行安装操作了，使用的命令如图 1.66 所示。

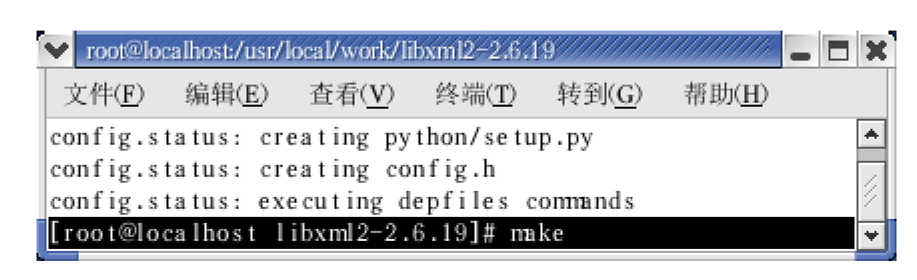


图 1.65　编译文件



图 1.66　执行安装

设计过程

本书使用的版本是 php-5.0.0.tar.gz。下面来看具体的安装步骤。

（1）将 php-5.0.0.tar.gz 解压到指定的文件夹下，通过命令进行配置，使用的操作命令如图 1.67 所示。

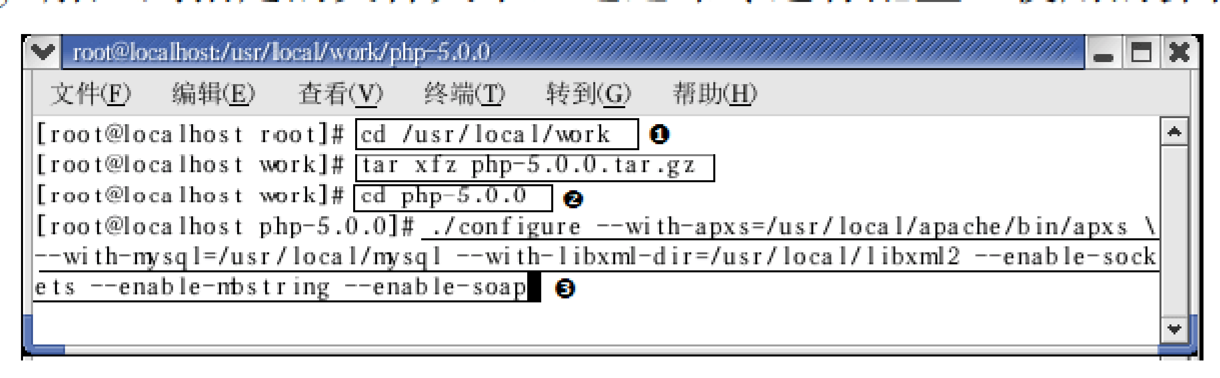
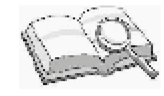


图 1.67　安装 PHP



代码导读

❶ 进入/usr/local/work 文件夹（php-5.0.0.tar.gz 压缩包存储的位置）。
❷ 解压 php-5.0.0.tar.gz。
❸ 配置文件。

（2）完成解压和配置操作后，对文件进行编译，使用的命令如下：

```
make
```

（3）编译完成后，执行安装操作，使用的命令如下：

```
make install
```

（4）最后，在命令中将出现如图 1.68 所示的内容，说明 PHP 安装成功。

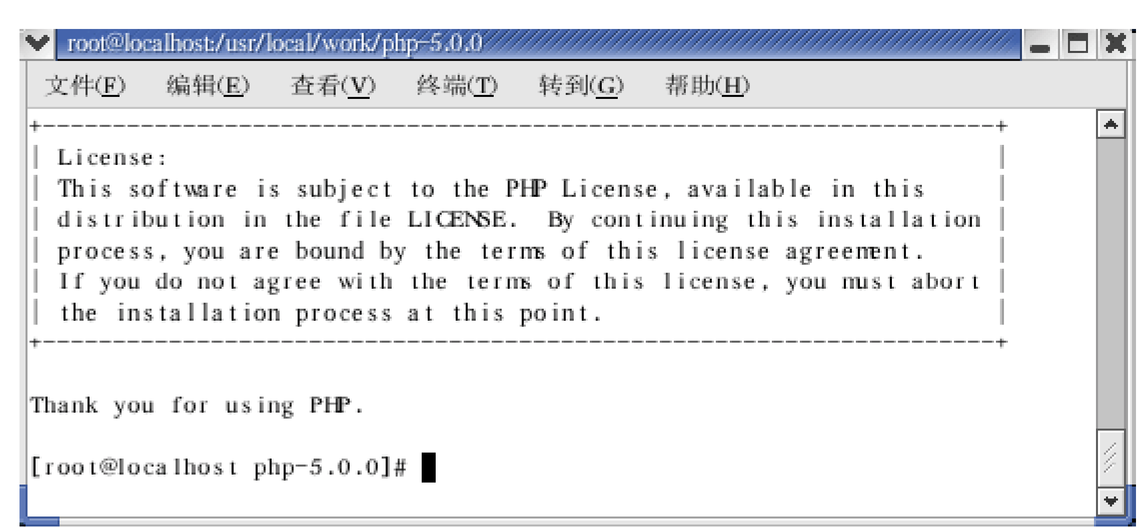


图 1.68　PHP 安装成功

秘笈心法

心法领悟 014：在 Linux 下创建 php.ini。

在 Linux 操作系统中，PHP 的配置文件仍然是 php.ini，其创建方法如下：

在 PHP 文件夹下找到 php.ini-dist 或者 php.ini-recommended，将其复制到/usr/local/lib 目录下，并重命名为 php.ini。

```
cp php.ini-dist /usr/local/lib/php.ini
```

实例 015　第 4 个 PHP 程序

光盘位置：光盘\MR\01\015

中级

趣味指数：★★★☆

实例说明

在 Linux 系统下搭建完成 PHP 开发环境后，编写一个小实例，测试环境是否搭建成功。在 Mozilla 浏览器的地址栏中输入 http://localhost/test.php，运行结果如图 1.69 所示。

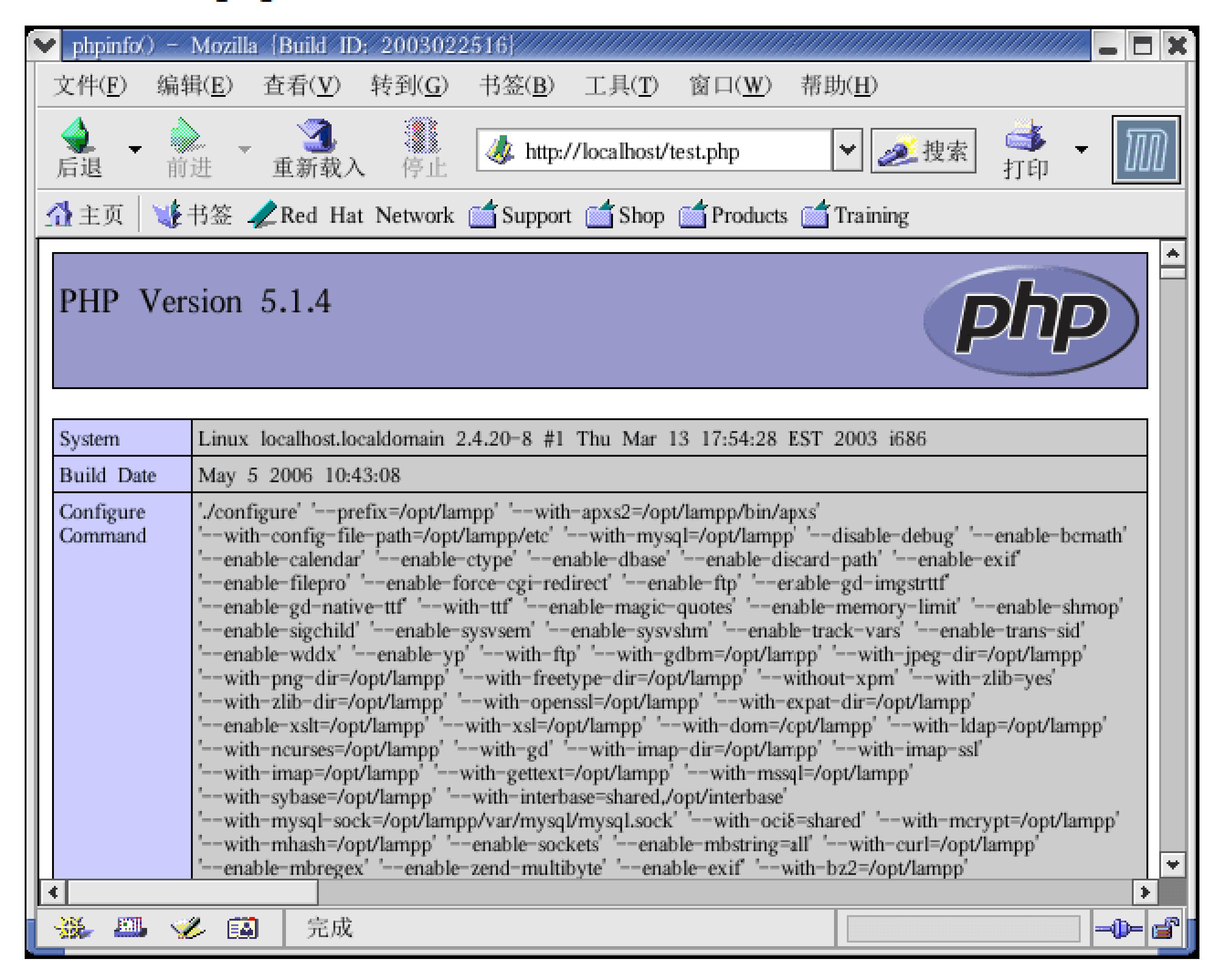


图 1.69　测试运行结果

关键技术

在 Linux 操作系统下，通过 gedit 工具编写 PHP 脚本。

设计过程

通过 gedit 工具编写一个 PHP 脚本 test.php，并存储于 Apache 的 htdocs 文件夹下。test.php 的代码如图 1.70 所示。

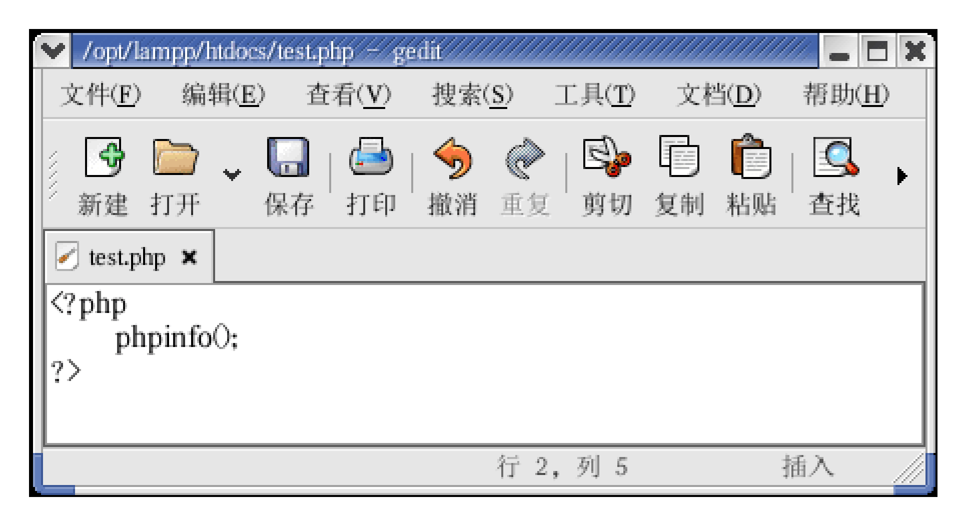
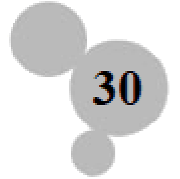


图 1.70　test.php 文件的内容



秘笈心法

心法领悟 015：Linux 下安装 XAMPP 集成环境。

本实例介绍的是在 Linux 操作系统下独立搭建 PHP 的开发环境，读者也可以使用 XAMPP 的 xampp-linux-1.7.3a.tar.gz 版本直接在 Linux 操作系统下搭建集成的开发环境。

1.5　XAMPP——Linux 版 PHP 集成化安装包

前面已经介绍过如何在 Windows 操作系统下应用 XAMPP 集成化安装包配置 PHP 的开发环境，这里将介绍如何在 Linux 操作系统下通过 XAMPP 配置 PHP 的开发环境以及其基本应用。

实例 016	XAMPP——Linux 下 PHP 开发环境的集成化 光盘位置：光盘\MR\01\016	中级 趣味指数：★★★☆

实例说明

在 Linux 操作系统下，也可以使用集成的软件来配置 PHP 开发环境，它就是 XAMPP 的 Linux 版。配置成功后，在 Mozilla 浏览器的地址栏中输入 http://127.0.0.1/index.php，运行结果如图 1.71 所示。

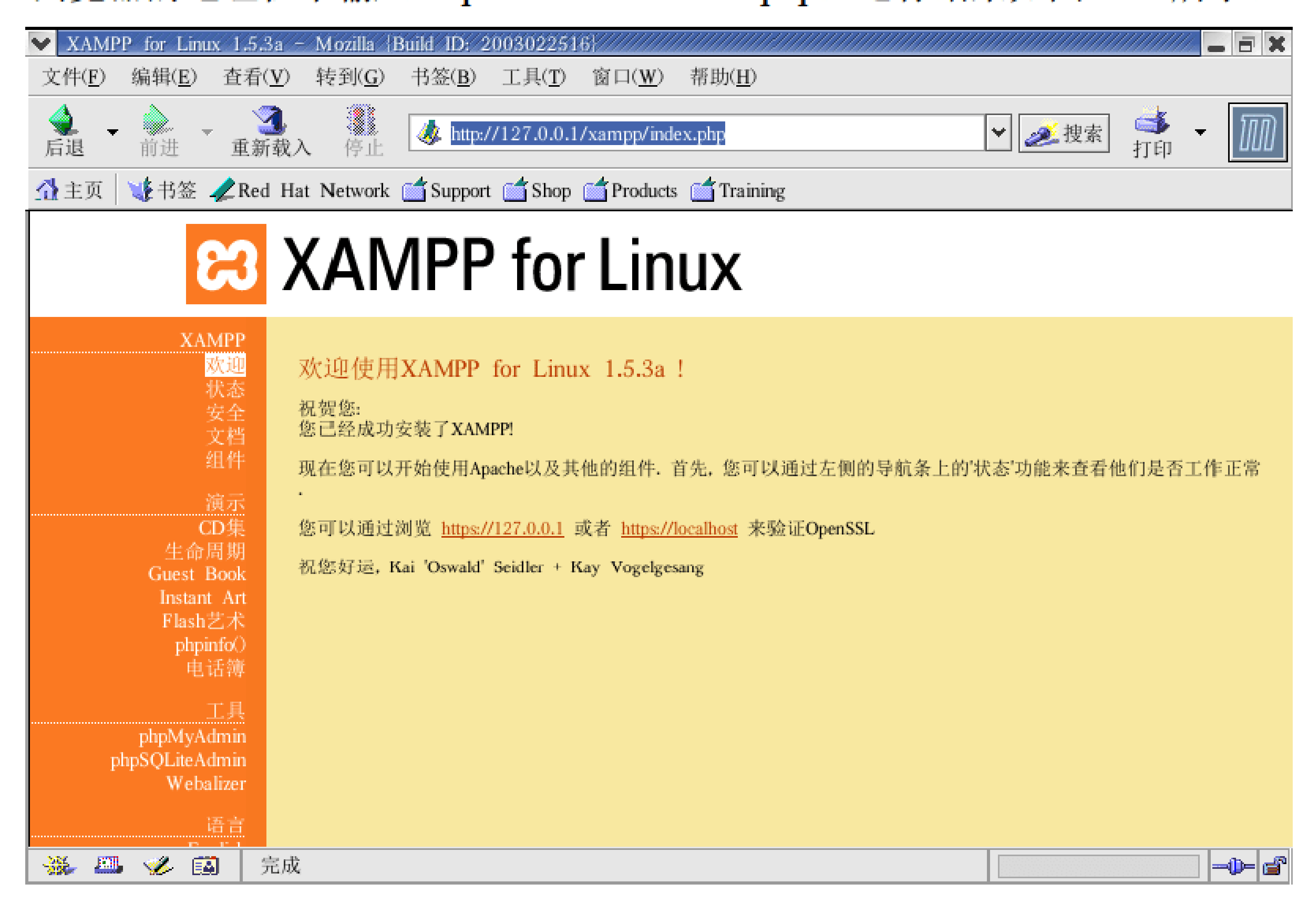


图 1.71　Linux 版的 XAMPP

关键技术

在 Linux 操作系统下安装 XAMPP，必须在 Linux “主菜单”/“系统工具”下的 “终端”命令中通过指令来完成，切记不要使用任何微软操作系统下的工具来进行操作。

设计过程

Linux 下安装 XAMPP 的步骤如下：

（1）在 Linux 操作系统下，选择“主菜单”/“系统工具”/“终端”命令。
（2）在命令模式下，首先进入系统的根目录。
（3）通过 mkdir 命令在根目录下创建一个 opt 目录。
（4）通过 tar xvfz 命令将 xampp 解压缩到 opt 目录下。
（5）按 Enter 键，执行 XAMPP 的解压缩，直到安装成功，其具体使用的命令如图 1.72 所示。
（6）安装成功后，查看/opt/lampp 目录，如图 1.73 所示。

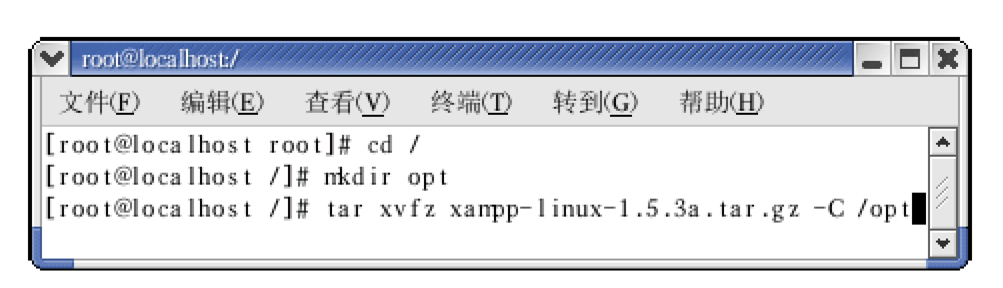


图 1.72　XAMPP 的安装

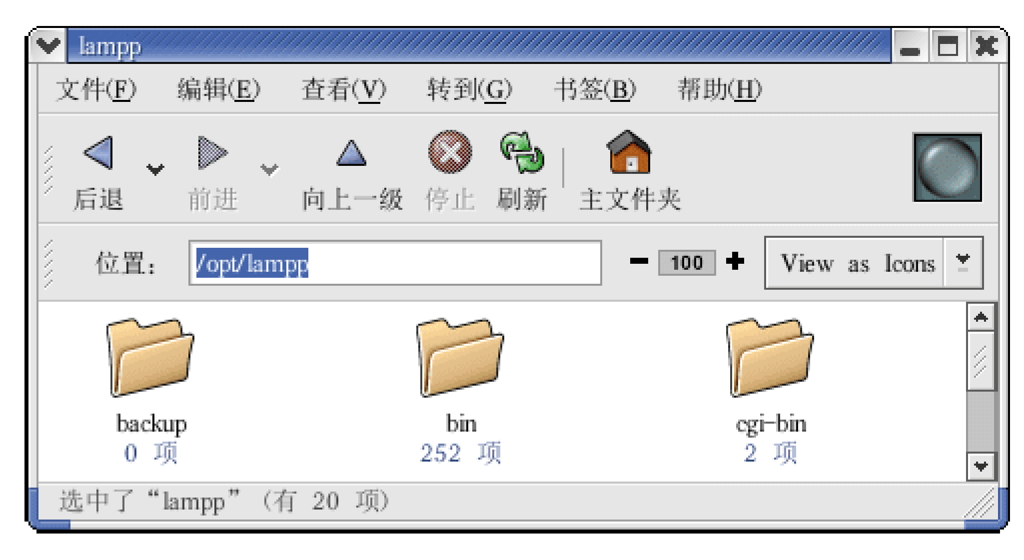


图 1.73　查看安装的内容

秘笈心法

心法领悟 016：Linux 下安装 XAMPP 的注意事项。

在 Linux 操作系统下执行 XAMPP 的解压缩时，必须进入系统的根目录下，并在根目录下创建一个子目录，然后才可以执行解压缩的命令，否则将提示找不到指定的目录或者文件。



实例说明

在实例 016 中，已经介绍了如何在 Linux 操作系统下安装 XAMPP，本实例我们将介绍在 Linux 操作系统下如何启动 XAMPP，其使用的命令如图 1.74 所示。



图 1.74　启动 XAMPP

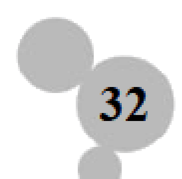
关键技术

在 Linux 操作系统下启动 XAMPP，同样要进入 Linux “主菜单”/“系统工具”下的 “终端”命令中，所使用的命令如下：

```
/opt/lampp/lampp start
```

设计过程

Linux 下启动 XAMPP 的步骤如下：



（1）在 Linux 操作系统下，选择“主菜单” / “系统工具” / “终端”命令。
（2）直接输入命令/opt/lampp/lampp start。
（3）按 Enter 键，执行 XAMPP 的启动操作。命令运行结果如图 1.75 所示。



图 1.75　XAMPP 启动成功

秘笈心法

心法领悟 017：停止 XAMPP 的命令。
Linux 下停止 XAMPP 的命令如下：

```
/opt/lampp/lampp stop
```

实例 018　设置 MySQL 数据库 root 用户的密码

光盘位置：光盘\MR\01\018

中级　趣味指数：★★★☆

实例说明

在 XAMPP 中，默认情况下通过 phpMyAdmin 以 root 用户的身份登录 MySQL 数据库是不需要密码的，为了确保数据库的安全，应该给 root 用户设置一个密码。这就是本实例要介绍的内容——如何为 root 用户设置密码。

关键技术

在 Linux 操作系统下为 MySQL 数据库的 root 用户设置密码使用的命令如下：

```
/opt/lampp/bin/mysqladmin –u root password "newpassword"
```

设计过程

Linux 下设置 MySQL 数据库 root 用户密码的步骤如下：
（1）在 Linux 操作系统下，选择“主菜单” / “系统工具” / “终端”命令。
（2）直接输入命令/opt/lampp/bin/mysqladmin –u root password "111"，这里设置 root 用户的密码为“111”。
（3）按 Enter 键，执行该命令，其运行结果如图 1.76 所示。



图 1.76　设置 root 用户的密码

秘笈心法

心法领悟 018：卸载 XAMPP 命令。
Linux 下卸载 XAMPP 的命令如下：

```
rm –rf /opt/lampp
```

实例 019 第 5 个 PHP 程序

光盘位置：光盘\MR\01\019

中级

趣味指数：★★★★☆

实例说明

Linux 系统下 PHP 的集成开发环境 XAMPP 已经配置成功，下面编写一个小实例，应用 phpinfo()函数输出 PHP 的配置信息，通过这个实例熟悉一下在 Linux 下 XAMPP 的运用。在 Mozilla 浏览器的地址栏中输入 http://127.0.0.1/text/index.php，运行结果如图 1.77 所示。

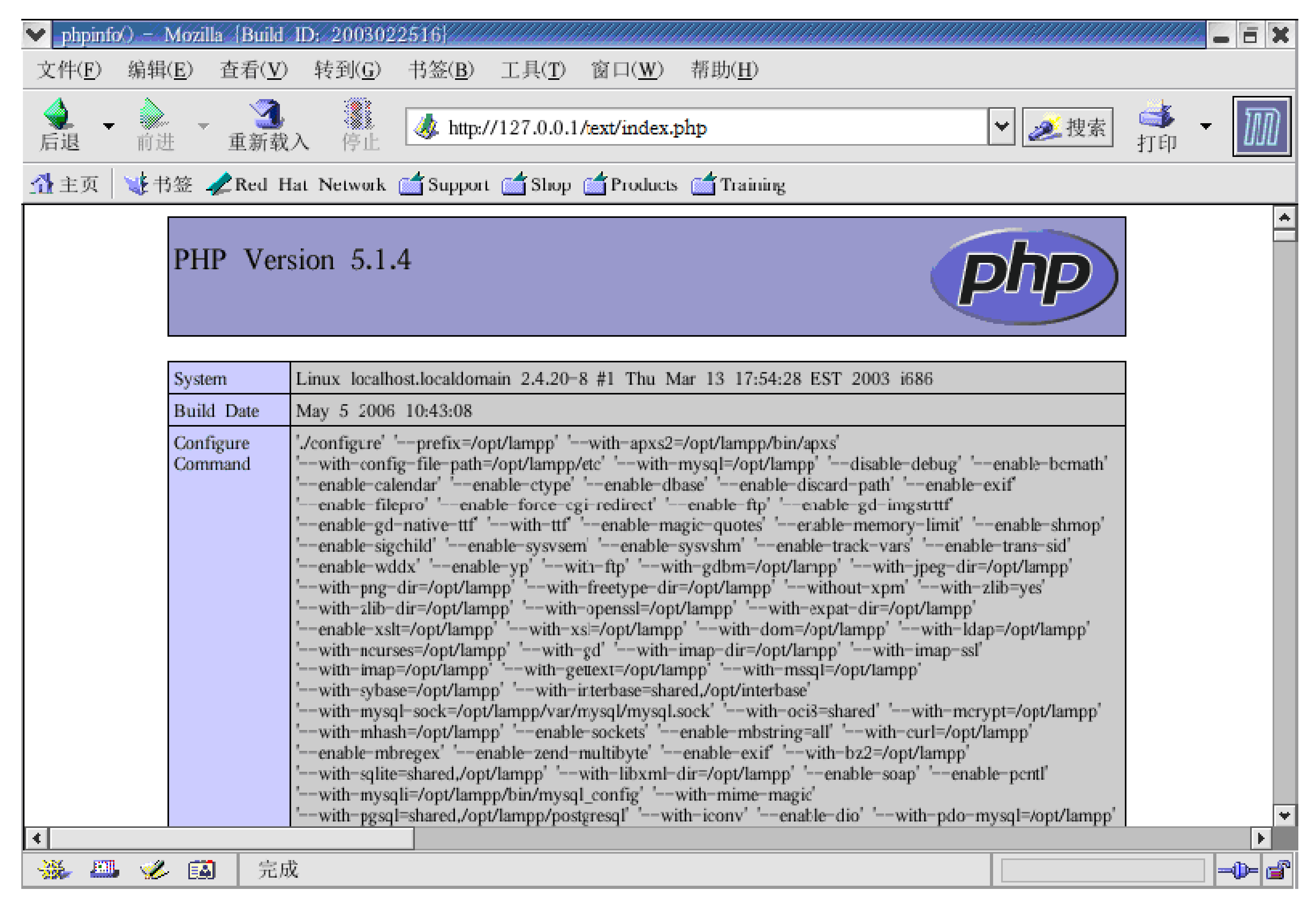


图 1.77 输出 PHP 的配置信息

关键技术

在 Linux 操作系统下运用 XAMPP 开发 PHP 程序，首先要确立 XAMPP 的程序运行文件夹，即明确创建的程序要存储在哪个文件夹中才可以运行。XAMPP 中的重要档案和目录如下：

- ☑ /opt/lampp/bin/：XAMPP 指令的根目录。例如，/opt/lampp/bin/mysql 用来执行 MySQL。
- ☑ /opt/lampp/htdocs/：Apache 文件根目录。
- ☑ /opt/lampp/etc/httpd.conf：Apache 设定档案。
- ☑ /opt/lampp/etc/my.cnf：MySQL 设定档案。
- ☑ /opt/lampp/etc/php.ini：PHP 设定档案。
- ☑ /opt/lampp/phpmyadmin/config.inc.php：phpMyAdmin 设定档案。

设计过程

在 XAMPP 中开发 PHP 程序的操作步骤如下：

（1）首先通过 gedit 工具编写一个 PHP 脚本，应用 phpinfo()函数输出 PHP 的配置信息，如图 1.78 所示。

（2）选择图 1.78 中的“文件”/“另存为”命令，将该文件存储于 XAMPP 的 htdocs/目录下，并且命名为 index.php，如图 1.79 所示。



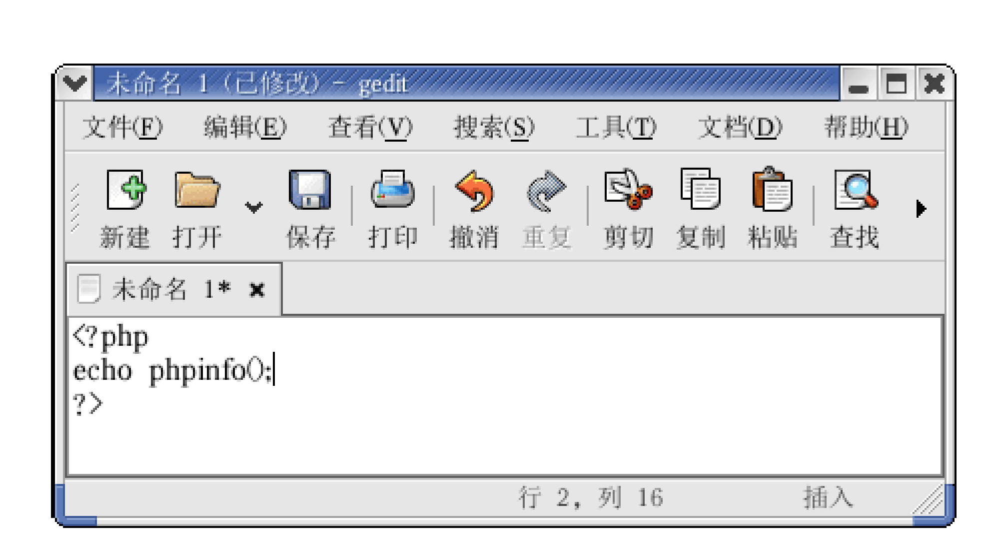


图 1.78　编辑 PHP 脚本文件

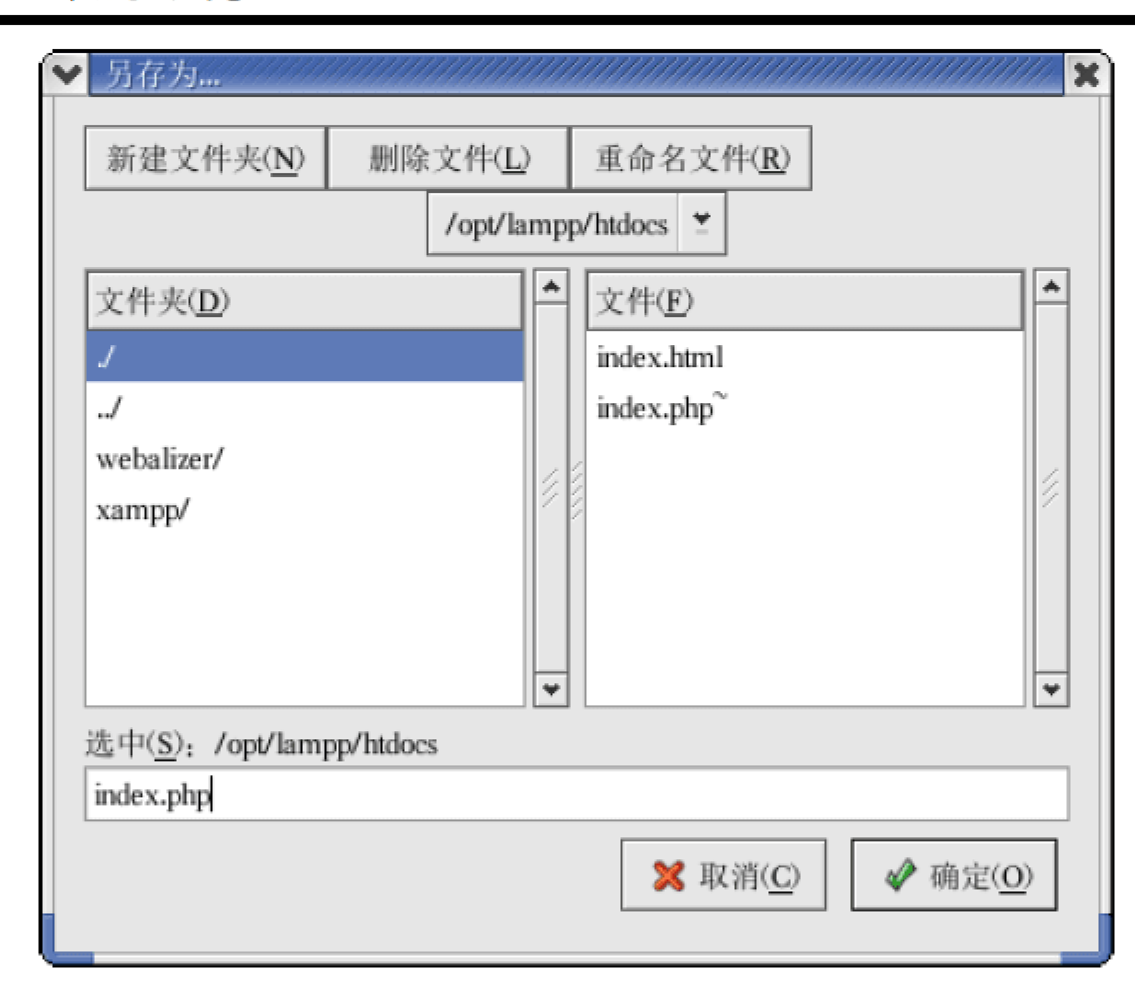


图 1.79　完成 PHP 文件的存储

（3）在 Mozilla 浏览器的地址栏中输入 http://127.0.0.1/text/index.php，运行结果如图 1.77 所示。

秘笈心法

心法领悟 019：Linux 下修改 MySQL 默认字符集。

在 Linux 下修改 MySQL 默认字符集为 utf8，操作的文件是 my.cnf，操作的内容如下：

```
[client]
default_character_set=utf8
[mysqld]
# default-character-set=utf8
character-set-server = utf8
collation-server = utf8_general_ci
[mysql]
default_character_set=utf8
```

1.6　Dreamweaver 开发工具

Dreamweaver 是 Macromedia 公司开发的 Web 站点和应用程序的专业开发工具，它将可视布局工具、应用程序开发功能和代码编辑组合在一起。

实例 020	Dreamweaver 中编码格式的选择 光盘位置：光盘\MR\01\020	中级 趣味指数：★★★☆

实例说明

应用 Dreamweaver 开发网站，首先必须考虑网页的编码格式的选择。如果网页的编码格式有局限性，那么将导致网站在一些特定的情况下运行时出现乱码，将不利于网站程序的后期更新和维护。例如，如果网站在编写时使用的是 GB2312 编码，当程序在繁体操作系统中运行时就会出现乱码，结果如图 1.80 所示。

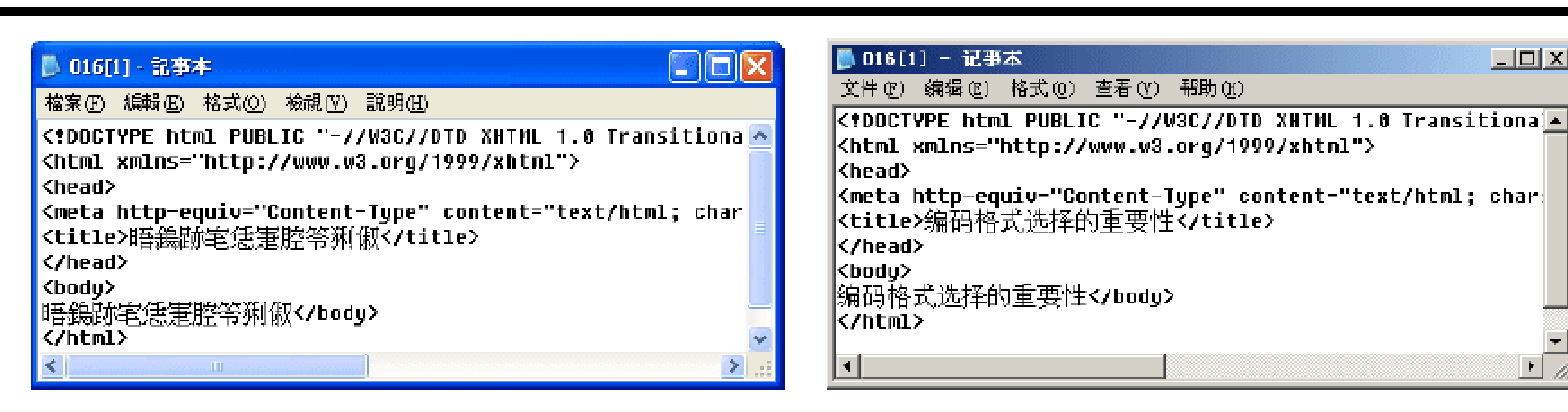


图 1.80　繁体操作系统下查看 GB2312 编码格式文件与源文件对比

关键技术

对于网站的开发，编码格式的选择很重要。如果使用 Dreamweaver 开发网站，那么可以选择“编辑”/“首选参数”/“新建文档”/“默认编码”命令设置新建文件的编码格式。这样就不必再为创建的每个文件的编码而担心，因为它们的编码格式是统一的。

设计过程

统一 Dreamweaver 创建文件编码格式的方法如下：

（1）打开 Dreamweaver 开发工具，选择“编辑”/“首选参数”命令，弹出如图 1.81 所示对话框。

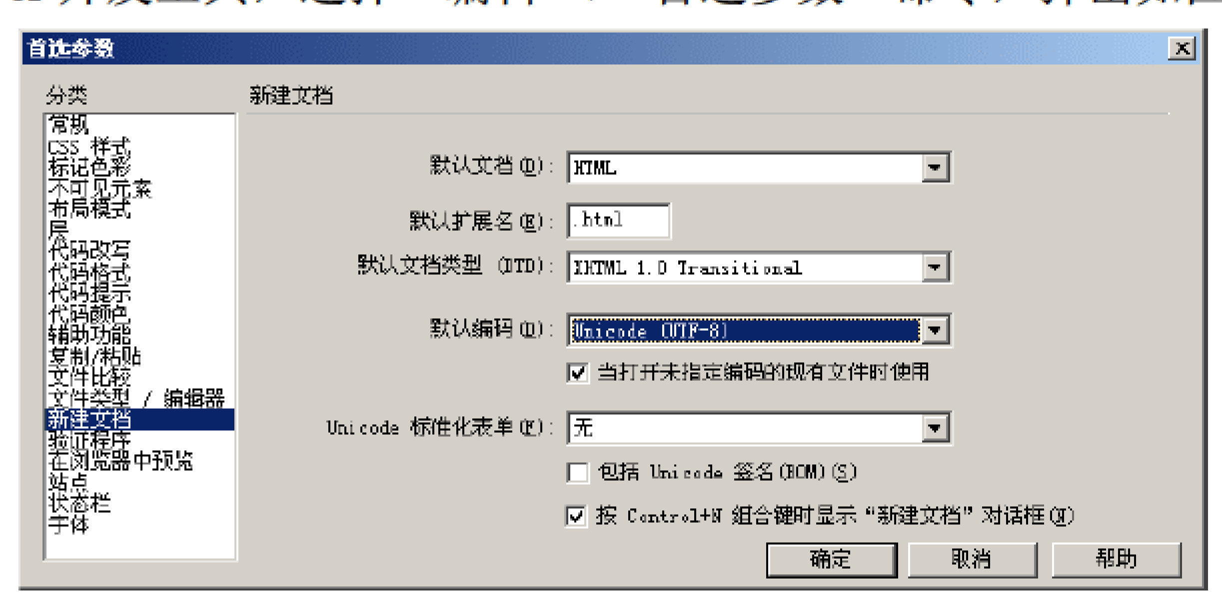


图 1.81　设置文件的编码格式

（2）在其中指定默认编码，最后单击“确定”按钮。

秘笈心法

心法领悟 020：在 Dreamweaver 中设置编码。

在 Dreamweaver 的“首选参数”设置中，不但可以设置默认编码的格式，还可以进行其他设置，如 CSS 样式、站点和字体等。

对于文件编码格式的选择，强烈建议读者使用 UTF-8，因为这样可以与国际接轨，如果单纯地使用 GB2312 编码，那么一旦程序要更改编码格式，网页将出现乱码。如果使用 UTF-8 编码格式，就不会出现任何问题，因为这个编码格式是通用的。



实例说明

实例 020 讲解了如何在 Dreamweaver 中设置编码格式，下面介绍一下 Dreamweaver 的具体应用——在 Dreamweaver 中创建表格。运行结果如图 1.82 所示。

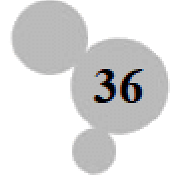




图 1.82　在 Dreamweaver 中创建表格

关键技术

在 Dreamweaver 中创建表格，有两种方法：

第一种，选择菜单栏中的“插入”/“表格”命令，在弹出的“表格”对话框中完成表格的创建操作。

第二种，直接在“常用”工具栏中单击“表格”按钮，在弹出的“表格”对话框中完成表格的创建操作。

设计过程

在 Dreamweaver 中创建表格的操作步骤如下：

（1）打开 Dreamweaver 开发工具，选择“文件”/“新建”命令，在弹出的“新建文档”对话框中创建一个“动态页”/PHP 文件，最后单击“创建”按钮，完成动态 PHP 文件的创建，如图 1.83 所示。

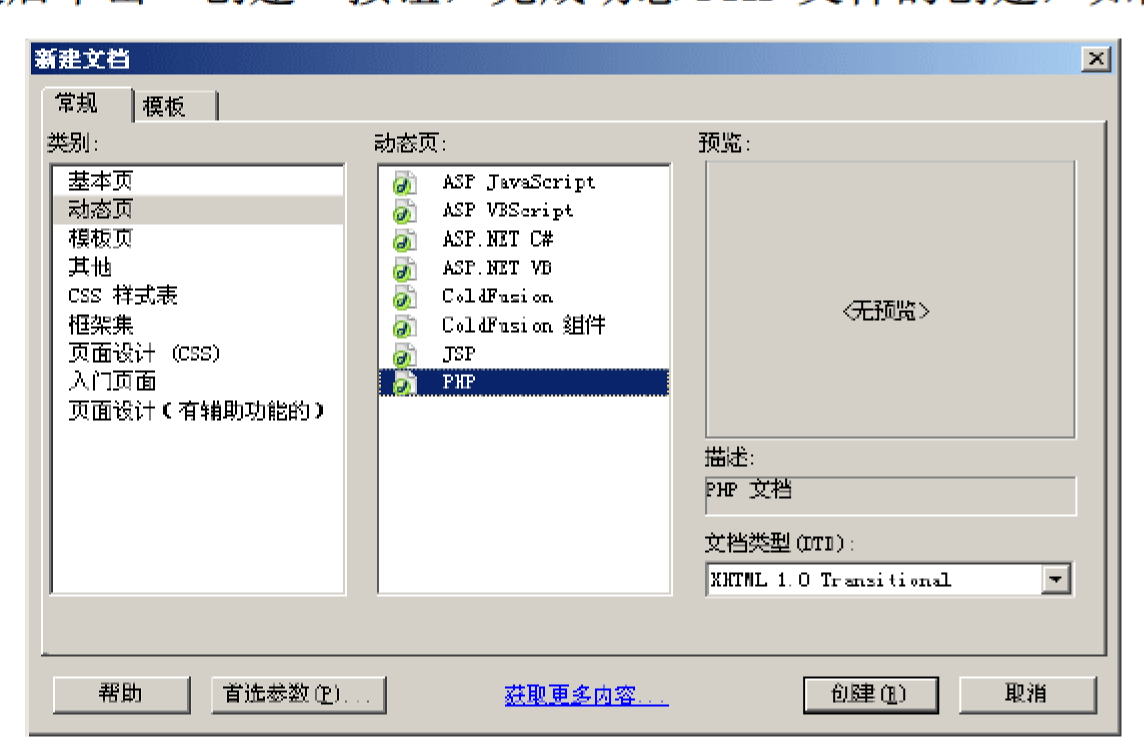


图 1.83　创建 PHP 动态文件

（2）如图 1.84 所示，在创建的 Untitled-1 文件中，首先切换到设计模式下，然后在工具栏中选择“常用”选项，单击“表格”按钮，在弹出的“表格”对话框中完成表格的创建。

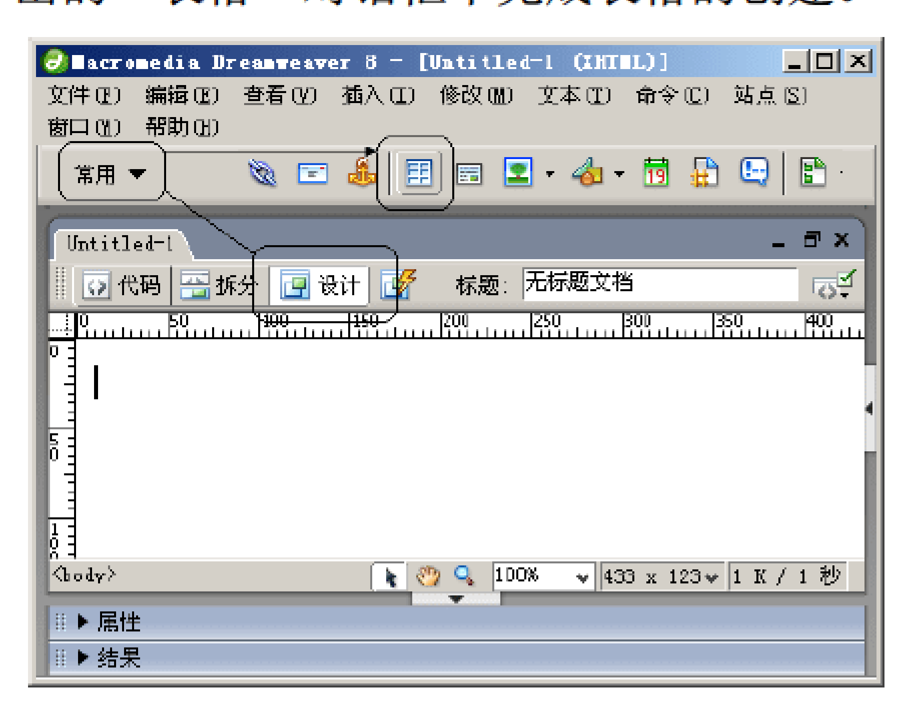


图 1.84　在设计模式下添加表格

（3）“表格”对话框如图 1.85 所示，在其中可以设置表格的行数、列数、表格宽度、边框粗细、单元格

边距和单元格间距等。设置完成后单击“确定”按钮，完成表格的创建。



图 1.85 “表格”对话框

（4）表格创建完成后，将在设计界面中输出如图 1.86 所示的内容。

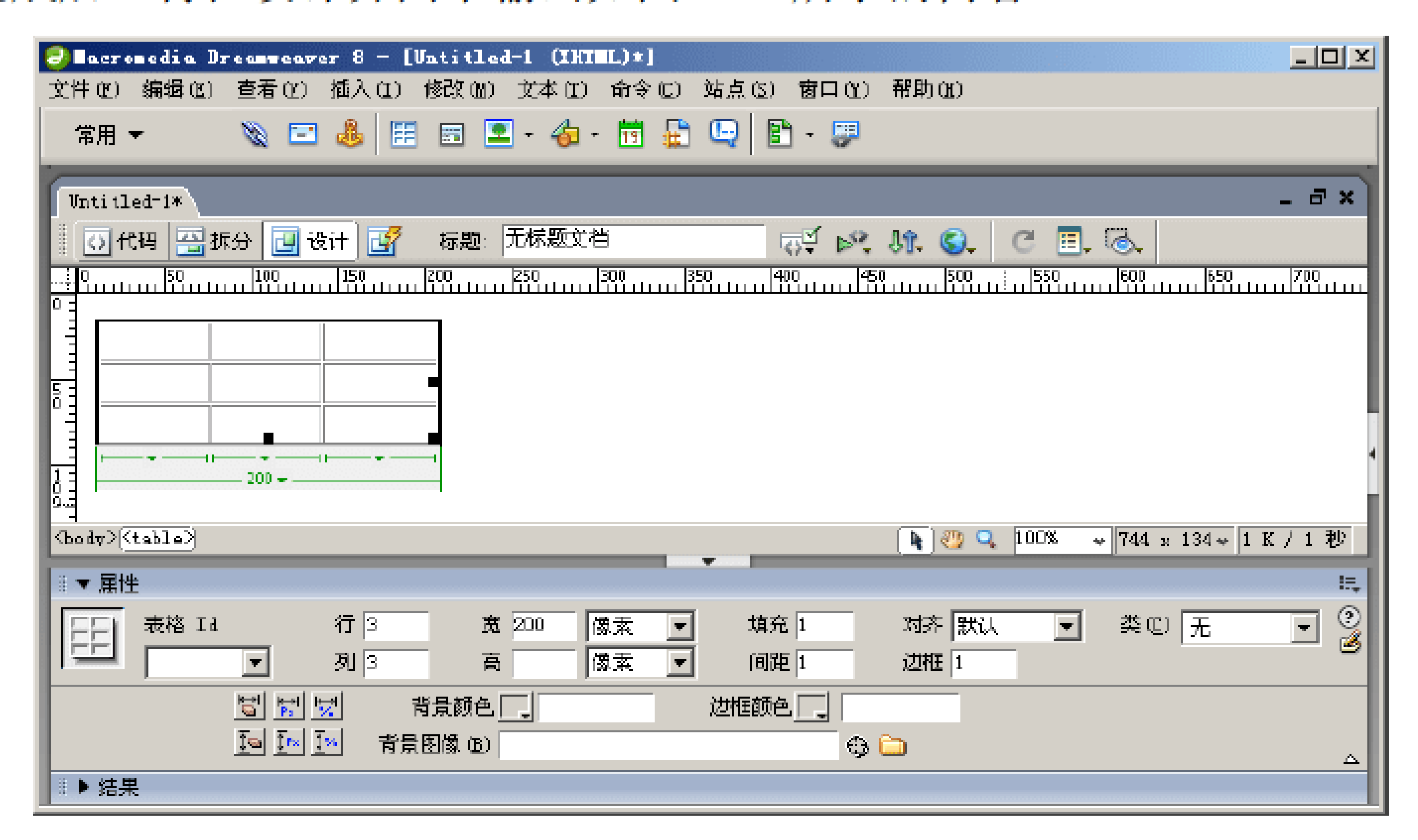


图 1.86 创建完成的表格

（5）选择“文件”/“保存”命令，将其存储于 Apache 服务器指定的文件夹下，命名为 index.php。运行结果请参考实例说明。

秘笈心法

心法领悟 021：操作表格的小技巧。

在图 1.86 所示的表格设计界面中，表格处于选中状态时，可以设置其属性，包括行数、列数、表格宽度、边框粗细、单元格边距、单元格间距、背景颜色、边框颜色和背景图像等。



实例说明

实例 021 讲解了如何在 Dreamweaver 中创建表格，本实例将讲解如何在表格中插入图片。本实例的运行结果如图 1.87 所示。





图 1.87　表格中插入宠物图片

关键技术

在 Dreamweaver 中，向创建的表格中插入图片时，必须注意图片的存储位置，最好的方式就是先将图片存储到实例的根目录下，这样在实例运行过程中，不会因为机器的更换而找不到图片。

如果没有将图片存储到实例的根目录下，在执行图片的插入操作过程中，将给出一个如图 1.88 所示的提示信息，提示是否将图片存储到实例的根目录下。

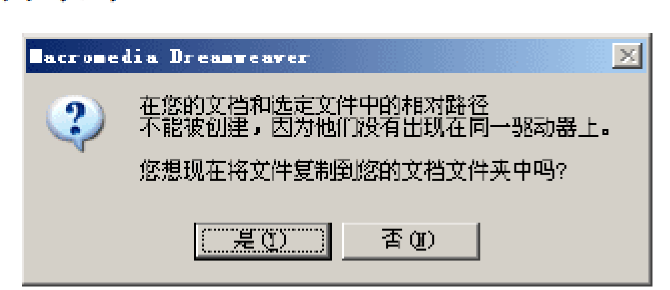


图 1.88　插入不在同一驱动器上的文件时给出提示

当单击“是”按钮时，即可将图片复制到实例的根目录下。但是，这种做法有一个弊端，就是文件的存储没有规范性。试想一下，如果网页中需要插入多个图片，程序的根目录将会怎样？所以最好的方法是先将要插入的图片存储到实例根目录下的一个指定文件夹中，这样做比较规范，也容易管理，一般都将这个图片存储文件夹命名为 images。

如果在图 1.88 的提示信息中，单击“否”按钮，虽然图片仍然可以插入到表格中，但在查看网页代码时会发现，img 标签中 src 属性指定的图片文件路径是本机的绝对路径（file:///E|/Program Files/9.gif），在本机中仍然可以运行，但要将本实例复制到其他机器上运行的话，在指定的路径下将找不到图片文件。

设计过程

在 Dreamweaver 中创建表格的操作步骤请参考实例 021，这里不再赘述，下面介绍如何在创建好的表格中插入图片。

（1）首先，将要使用的图片复制到本实例根目录下的 images 文件夹中。

（2）将光标定位到要插入图片的单元格中，选择 Dreamweaver 开发工具中的“插入”/“图像”命令，在弹出的“选择图像源文件”对话框中选择要插入的图片，如图 1.89 所示。



图 1.89　插入图片

（3）确定要插入的图片之后，单击“确定”按钮，即完成图片的插入操作。

（4）保存文件，在 IE 浏览器中输入网址，即可运行本实例，运行结果请参看实例说明。

秘笈心法

心法领悟 022：操作 Dreamweaver 小技巧。

在 Dreamweaver 的设计模式下，当选中插入的图片时，在开发工具的下方将出现图像的属性操作界面，如图 1.90 所示。通过这个属性操作界面可以设置图像的宽度、高度、链接以及是否添加地图等。

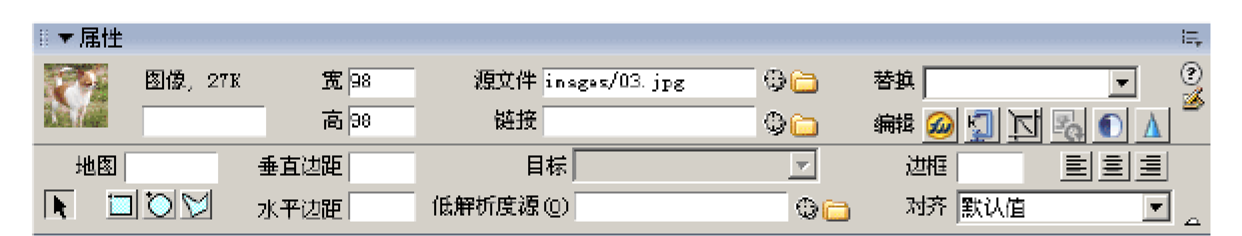


图 1.90　图像的属性操作界面

实例 023　Dreamweaver 中创建表单

光盘位置：光盘\MR\01\023

中级

趣味指数：★★★☆

实例说明

本实例讲解如何在 Dreamweaver 中创建表单。表单是网站与客户沟通的一座桥梁，通过它可以直接将客户的信息反馈给网站的管理者，达到企业与网站浏览者更好的交互效果。在本实例中将对如何在 Dreamweaver 中创建表单进行详细讲解，实例的运行结果如图 1.91 所示。

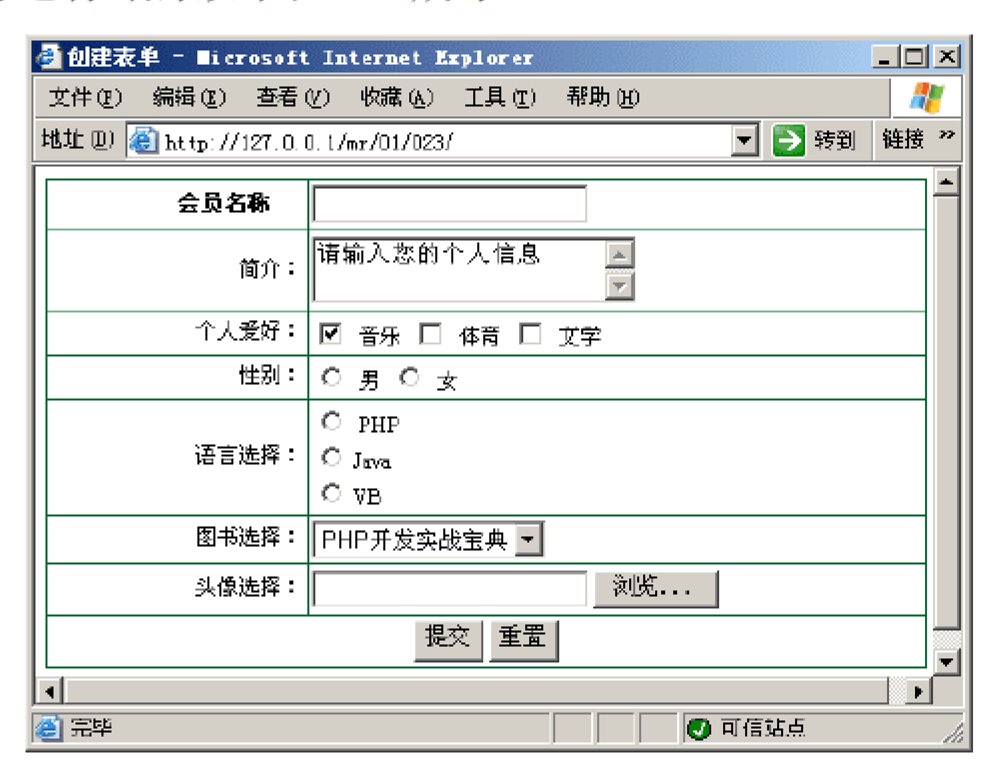


图 1.91　创建表单元素

关键技术

在 Dreamweaver 中创建表单，首先要在工具栏中选择“表单”选项，然后就可以通过不同的按钮创建不同的表单元素，其中每个按钮对应的功能如表 1.1 所示。

表 1.1　Dreamweaver 中的表单元素

图　　像	名　　称	说　　明
	表单	<form name="form2" method="post" action=""></form> name：表单的名称 method：表单提交的方法，包括 POST 和 GET 方法 action：表单提交的路径

续表

图　像	名　称	说　明
	文本字段	<input type="text" name="textfield"> type：应用表单的类型 name：文本框的名称
	隐藏域	<input type="hidden" name="ID" value=""> type：表单的类型，其中的 hidden 表示隐藏域 name：隐藏域的名称，可以自己定义 value：隐藏域的值，可以填写隐藏域的默认值
	文本区域	<textarea name="" cols="" rows="" id=""></textarea> <textarea>…</textarea>：文本域的标记 name：文本域的名称，例如其中的 test cols：表示文本域字符的宽度 rows：表示有多少行字符 初始值在<textarea></textarea>标记之间进行输入，例如其中的“欢迎大家访问我们的论坛”
	复选框	<input type="checkbox" name="checkbox" value="体育"> type：表单的类型，其中的 checkbox 表示复选框 name：复选框的名称，例如其中的 checkbox value：复选框提交的值，例如其中的“体育” checked：如果希望预先为用户选中某些选项，可以为这些选项加上 checked 参数 disabled：如果希望某一个选项失效，可以加上 disabled 参数
	单选按钮	<input type="radio" name="radiobutton" value="radiobutton" checked="checked" />男 <input type="radio" name="radiobutton" value="radiobutton" />女
	单选按钮组	<input type="radio" name="RadioGroup1" value="单选" />单选 <input type="radio" name="RadioGroup1" value="单选" />单选
	列表\菜单	<select name="select2"> <option selected="selected">默认值</option> <option value="对应值 1">列表值 1</option> <option value="对应值 2">列表值 2</option> … </select> name：指该<select>组件的名称 option：提供给用户选择的项目，其中的 value 是该选项所代表对应的选择值，可以省略
	跳转菜单	<select name="menu1" onchange="MM_jumpMenu('parent',this,0)"> <option>unnamed1</option> </select> 跳转菜单，通过表单实现指定网址之间的跳转
	图像域	<input type="image" name="imageField" src="images/QQ.gif" />在表单中插入图片
	文件域	<input type="file" name="file" />完成文件的提交
	按钮	<input type="submit" name="Submit" value="提交" />创建的提交按钮，如果 type 的值为 button，那么它表示一个普通的按钮，不具备提交的功能 <input type="reset" name="Submit2" value="重置" />创建的重置按钮
	标签	<label>标签</label>
	字段集	<fieldset><legend>字段集</legend></fieldset>

设计过程

在 Dreamweaver 中创建表单的操作步骤如下：

（1）新建一个动态 PHP 文件，切换到设计模式下。

（2）在工具栏中选择“表单”选项，如图 1.92 所示。

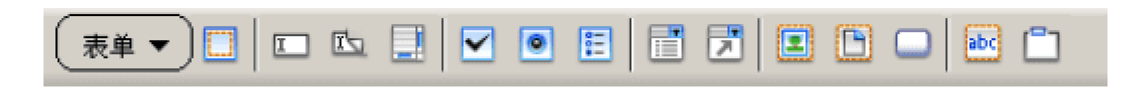


图 1.92　选择表单工具

（3）添加一个 form 表单，如图 1.93 所示。

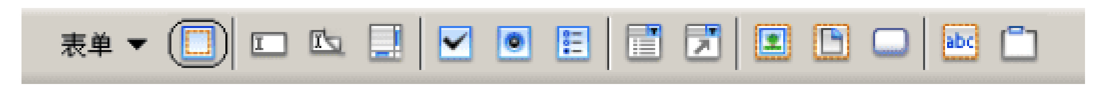


图 1.93　添加表单

（4）根据实际需要，添加不同的表单元素，并设置相应的名称和值，如图 1.94 所示。



图 1.94　添加表单元素

最终创建的表单元素设计效果如图 1.95 所示。

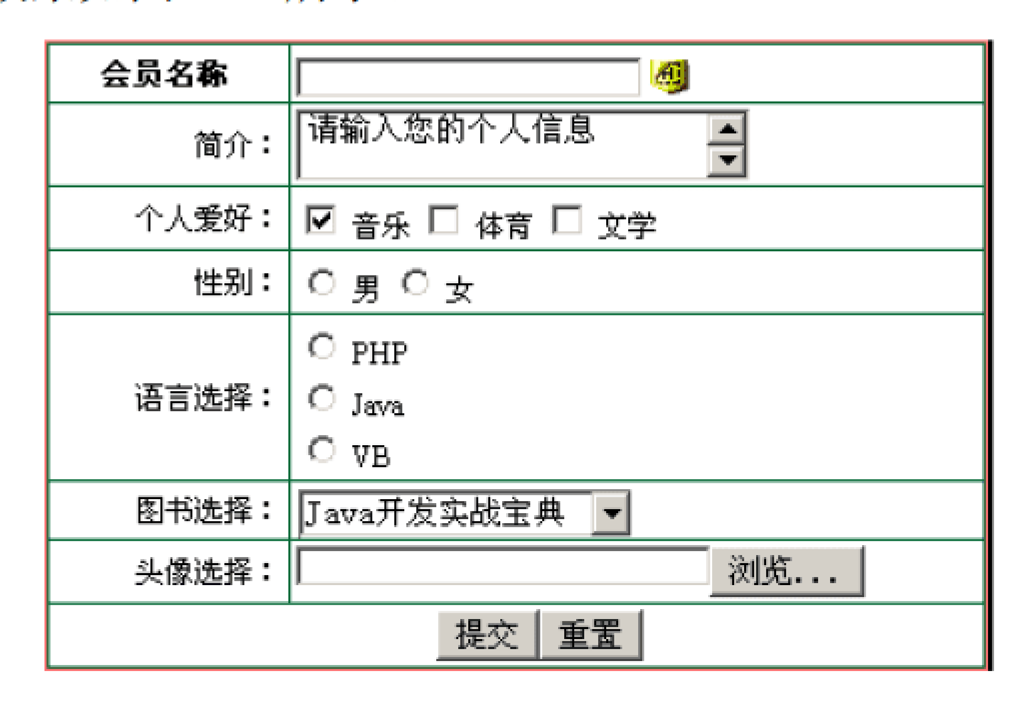


图 1.95　表单元素的设计效果

秘笈心法

心法领悟 023：通过表单传值时的注意事项。

在通过表单中的元素传递值的时候，一定要正确地书写表单元素的名称，其中不应该有空格存在；在获取表单元素的值时，表单元素的名称一定要与 form 中设置的名称相同，同时还要注意大小写的统一，否则将不能获取到表单元素的值。

实例 024　Dreamweaver 中创建和附加 CSS 样式

光盘位置：光盘\MR\01\024

中级

趣味指数：★★★☆

实例说明

本实例讲解如何在 Dreamweaver 中创建和附加 CSS 样式。通过 CSS 能够更好地对网页中的表格、文本和图像的样式进行控制，并且使其更易于维护和更新。本实例的运行结果如图 1.96 所示。



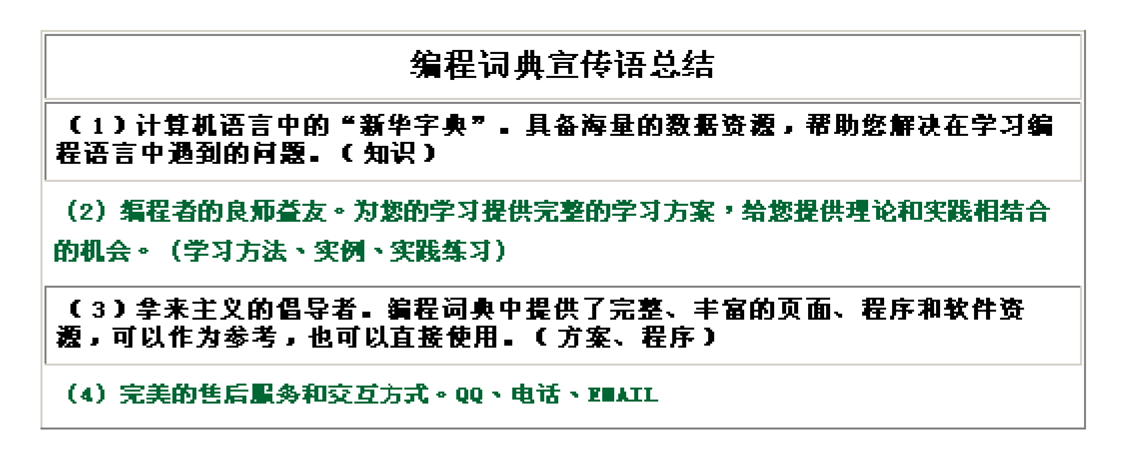


图 1.96　Dreamweaver 中创建和附加 CSS 样式

关键技术

在 Dreamweaver 中创建 CSS 样式，必须明确新建 CSS 样式的规则，如图 1.97 所示。

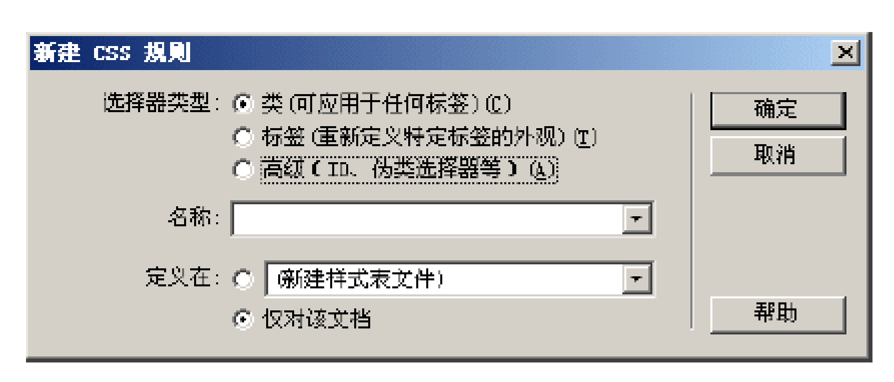


图 1.97　新建 CSS 规则

如果选中"（新建样式表文件）"单选按钮，那么定义的 CSS 存储于指定的文件夹下；如果选中"仅对该文档"单选按钮，那么定义的 CSS 存储于该文件中。

如果只是对单独的某个文件进行操作，建议将 CSS 定义到当前文件中，但是，如果是在网站中应用 CSS，那么应该将 CSS 样式定义到单独的文件中，并且以类的形式进行定义，因为这样便于对 CSS 样式的修改，使网站的样式更加统一，而且可以避免代码的冗余。

设计过程

创建一个名称为 styles 的样式，将其定义到新建样式表中，使用类的形式进行定义，其具体的操作步骤如下：

（1）在 Dreamweaver 的设计模式下右击，在弹出的快捷菜单中选择"CSS 样式"/"新建"命令，在弹出的"新建 CSS 规则"对话框中定义 CSS 样式的规则，如图 1.98 所示。

（2）CSS 样式的规则定义完成后，单击"确定"按钮，弹出"保存样式表文件为"对话框，定义 CSS 样式文件的存储位置，如图 1.99 所示。

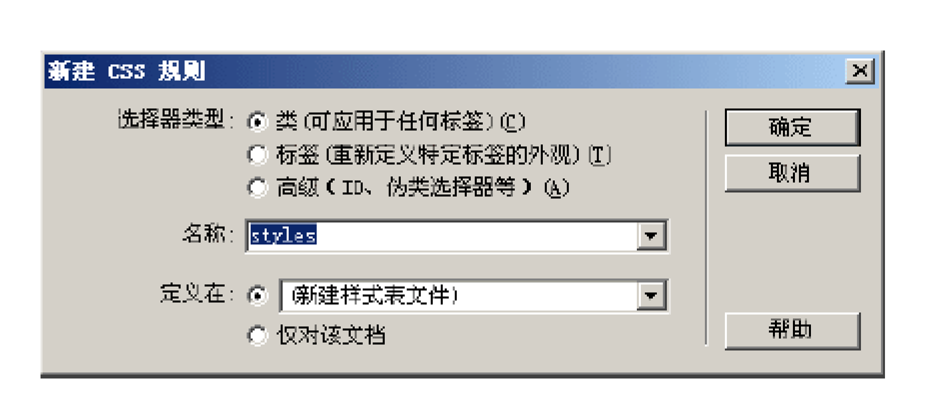


图 1.98　定义 CSS 样式的规则



图 1.99　定义 CSS 样式表的存储位置

（3）确定存储位置和名称后，单击"保存"按钮，将弹出 styles 样式具体的规则定义对话框，如图 1.100 所示。在其中，可以根据"分类"，对类型、背景、区块、方框、边框、列表、定位和扩展分别进行定义，单击"确定"按钮，styles 样式创建完成。

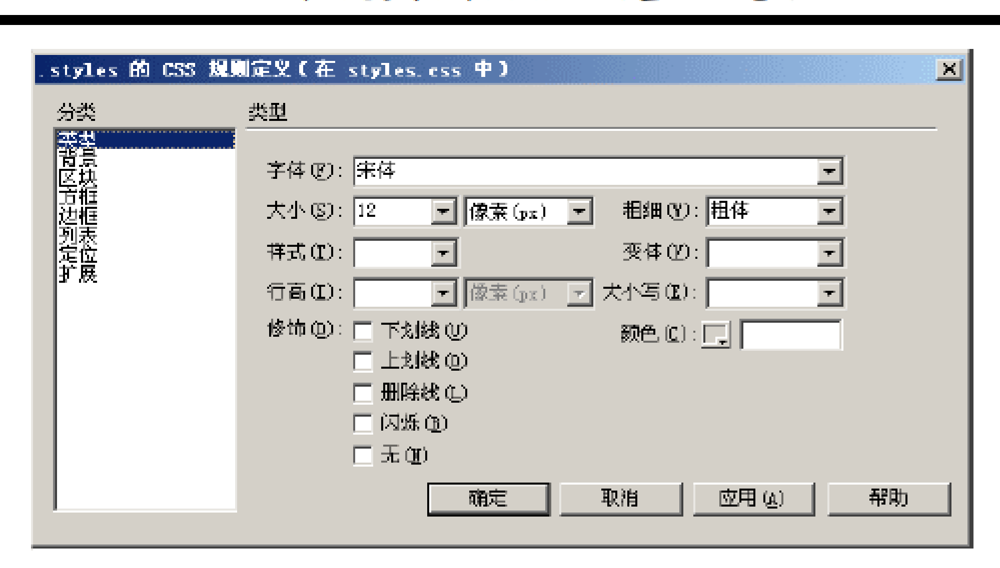


图 1.100　定义 styles 样式的规则

（4）在 Dreamweaver 的设计模式下，选中被操作的内容，右击，在弹出的快捷菜单中选择“CSS 样式”命令，即可选择指定的样式文件，如图 1.101 所示。

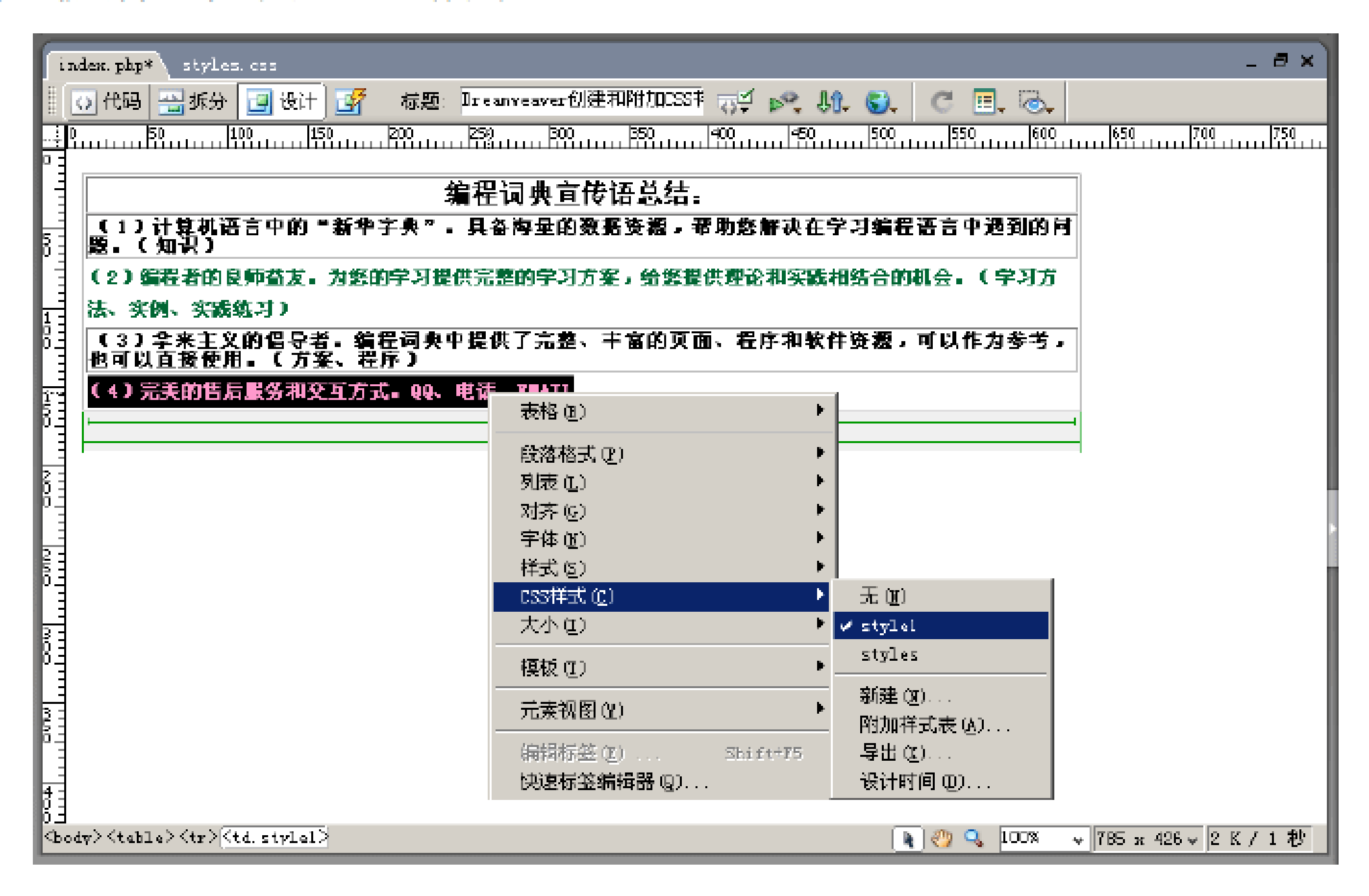


图 1.101　应用定义的 CSS 样式

技巧：附加 CSS 样式文件。在 Dreamweaver 的设计模式下右击，在弹出的快捷菜单中选择“CSS 样式”/“附加样式表”命令，在弹出的“链接外部样式表”对话框中可选择定义在指定文件夹下的 CSS 样式文件。如果选择“链接”，则直接通过 link 标签链接到 CSS 样式文件；如果选择导入，则通过 @import url 导入指定的 CSS 文件。确定附加的 CSS 样式文件后，单击“确定”按钮。

秘笈心法

心法领悟 024：在 Dreamweaver 中创建 CSS 样式的相关说明。

在 Dreamweaver 中创建的 CSS 样式，可以在代码模式下对其进行修改，也可以在设计模式下对其进行修改，如图 1.102 所示。

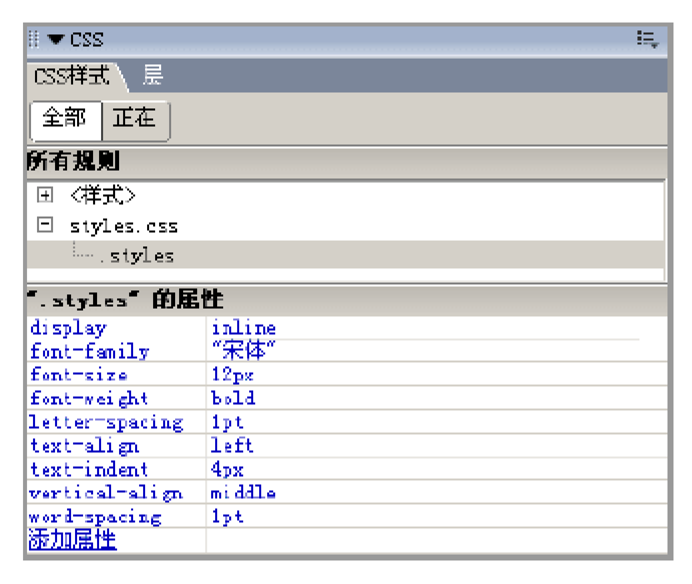


图 1.102　在设计模式下操作 CSS 样式



实例 025	Dreamweaver 控制弹出信息 光盘位置：光盘\MR\01\025	中级 趣味指数：★★★☆

实例说明

在 Dreamweaver 中不但可以创建表格、表单或者 CSS 样式等内容，而且可以通过行为，实现一些特殊的功能。例如，在本实例中定义一个“弹出信息”行为，当用户单击指定的图片时，将弹出一个提示对话框，如图 1.103 所示。



图 1.103　通过 Dreamweaver 直接创建的弹出信息

关键技术

在 Dreamweaver 中控制弹出信息的原理是自动创建 JavaScript 脚本文件，定义方法，然后通过 onclick 调用脚本中的方法，完成当鼠标单击指定的内容时弹出信息。

JavaScript 脚本中定义的方法如下：

```
<script type="text/JavaScript">
<!--
function MM_popupMsg(msg) { //v1.0
  alert(msg);
}
//-->
</script>
```

在表单的图像域中，通过 onclick 事件完成对脚本文件方法的调用，其代码如下：

```
<input name="imageField" type="image" onclick="MM_popupMsg('请输入完整的注册信息')" src="images/09.jpg" />
```

设计过程

在 Dreamweaver 中添加“弹出信息”行为的操作步骤如下：

（1）通过 Dreamweaver 新建一个 index.php 文件，创建一个用户注册页面。

（2）在 Dreamweaver 的设计模式下，选择用户注册的图像域，然后选择“窗口”/“行为”命令，在 Dreamweaver 操作界面的右侧将弹出如图 1.104 所示的内容。

（3）在图 1.104 所示的界面中，单击+按钮，在弹出的下拉菜单中选择“弹出信息”选项，将弹出如图 1.105 所示的“弹出信息”对话框，在“消息”文本框中定义弹出信息的内容，最后单击“确定”按钮，弹出信息的

行为创建完毕。



图 1.104　行为控制面板

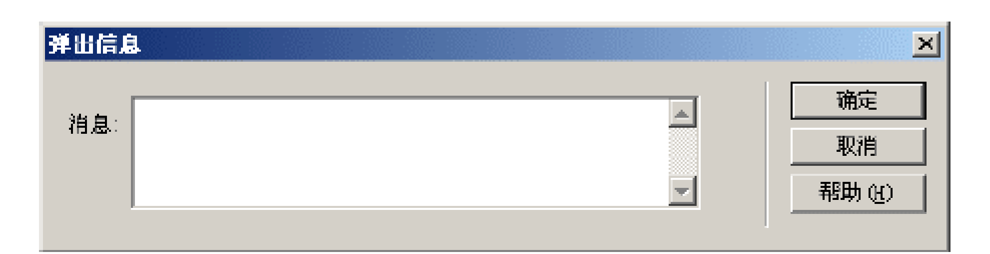


图 1.105　编辑弹出信息

秘笈心法

心法领悟 025：在 Dreamweaver 中实现一些 JavaScript 功能。

在 Dreamweaver 中，“弹出信息”只是众多行为中的一个，还可以通过“窗口”中的行为实现很多的功能，如图 1.106 所示。

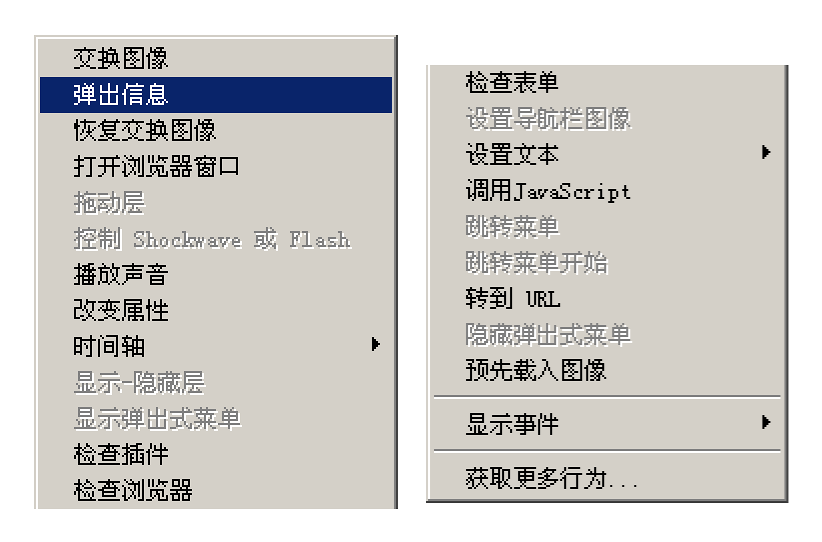


图 1.106　行为的功能展示

实例 026　Dreamweaver 控制浏览器的窗口

光盘位置：光盘\MR\01\026

中级

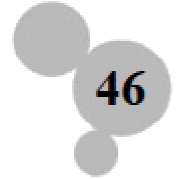
趣味指数：★★★☆

实例说明

本实例将介绍 Dreamweaver 中行为的另一种功能——控制打开浏览器的窗口。通过该功能可以对打开的浏览器窗口进行设置，包括窗口的高度、宽度以及是否包含导航工具栏、地址工具栏和菜单栏等，本实例的运行结果如图 1.107 所示。



图 1.107　Dreamweaver 控制浏览器的窗口



关键技术

控制打开浏览器的窗口同样是 Dreamweaver 中行为的一种功能，通过该功能可以控制网页窗口的打开模式。例如，网页窗口中导航条的使用、滚动条的使用、地址工具栏的应用以及网页窗口的大小等。

行为中“打开浏览器窗口”的操作界面如图 1.108 所示。

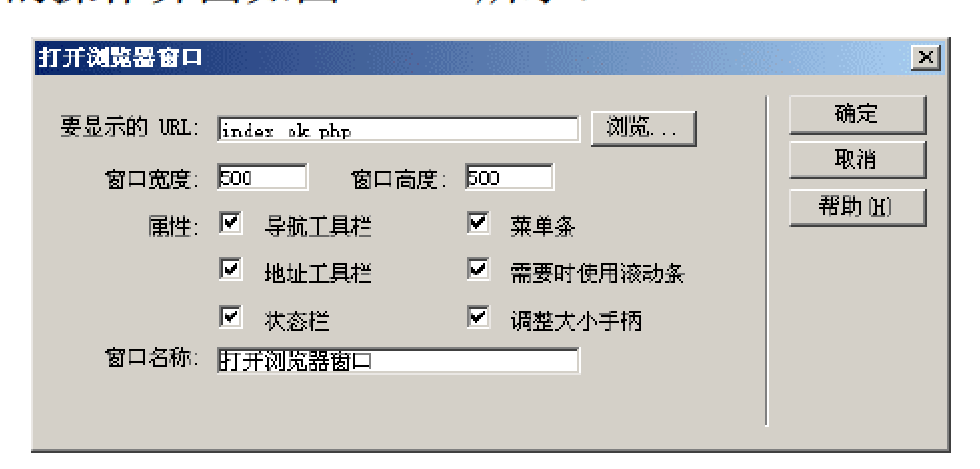


图 1.108　“打开浏览器窗口”对话框

设计过程

在 Dreamweaver 中添加“打开浏览器窗口”行为的操作步骤与添加“弹出信息”行为是相同的，唯一的区别是在单击按钮时，在弹出的下拉菜单中选择“打开浏览器窗口”命令，将弹出如图 1.108 所示的“打开浏览器窗口”对话框。

秘笈心法

心法领悟 026：在 Dreamweaver 中屏蔽标准按钮。

通过“打开浏览器窗口”行为，可以屏蔽浏览器窗口中的标准按钮，这个功能在某些程序中是非常有用的。例如，在一些需要屏蔽刷新功能的网页中，通过“打开浏览器窗口”行为来实现是比较不错的方法。

实例 027	通过 Dreamweaver 创建站点 光盘位置：光盘\MR\01\027	中级 趣味指数：★★★☆

实例说明

本实例讲解如何在 Dreamweaver 中创建站点，从而不必在 IE 浏览器中输入地址，只需按 F12 键，即可完成对所创建程序的浏览操作。

关键技术

在 Dreamweaver 中创建站点和配置测试服务器时，一定要注意将本地的 HTTP 地址与测试服务器中的 URL 前缀统一，即都指定到站点的根目录下。例如，指定 HTTP 的地址是 http://localhost/mr/01/，那么测试服务器的 URL 前缀也必须是 http://localhost/mr/01/或者 http://127.0.0.1/mr/01/。

设计过程

在 Dreamweaver 中创建站点的操作步骤如下：

（1）打开 Dreamweaver 开发工具，选择菜单栏中的“站点”/“新建站点”命令，在弹出的如图 1.109 所示对话框中添加站点名称、设置本地根文件夹、链接相对于“站点根目录”、设置 HTTP 地址。

（2）选择左侧的“测试服务器”选项，进入如图 1.110 所示的“测试服务器”对话框，选择服务器模型 PHP MySQL，访问为“本地/网络”，测试服务器文件夹为 F:\PkhPHP\www\，URL 前缀为 http://localhost/，最后单

击“确定”按钮。



图 1.109　定义 mr 站点

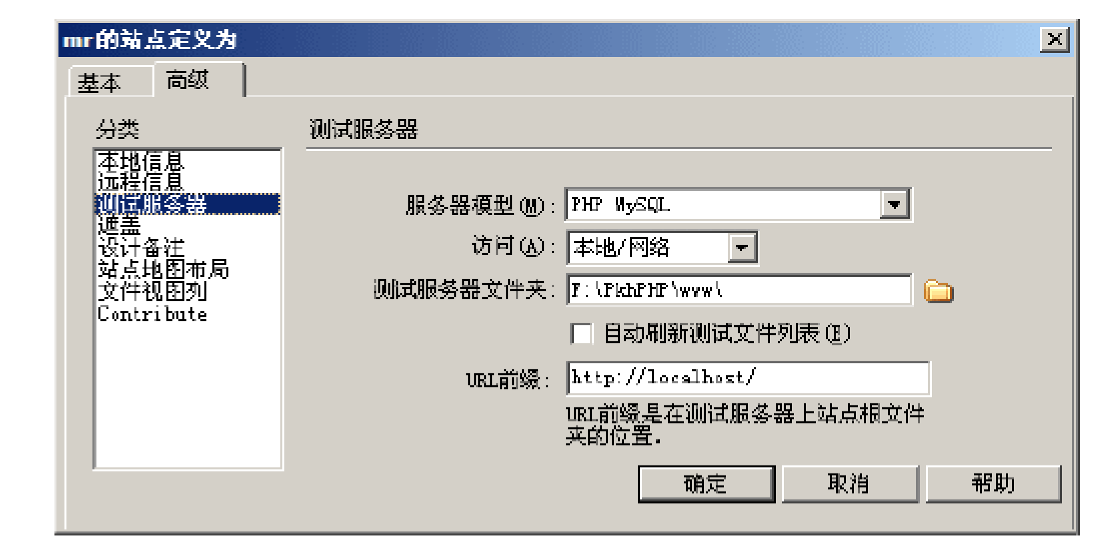


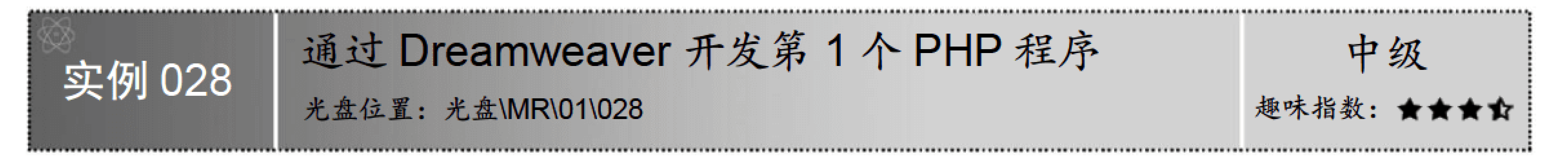
图 1.110　配置测试服务器

（3）mr 站点和测试服务器设置完毕后，就可以在 Dreamweaver 下直接使用 F12 键来浏览程序了。

秘笈心法

心法领悟 027：在 Dreamweaver 中创建站点的注意事项。

在 Dreamweaver 中创建站点和配置测试服务器时，如果本地的 HTTP 地址与测试服务器中的 URL 前缀不统一，那么就不能够通过 F12 键直接浏览程序。



实例说明

前面实例讲解了 Dreamweaver 自身功能的运用，在本实例中应用 Dreamweaver 开发一个最简单的 PHP 程序，即输出一段欢迎信息，其目的一是了解 PHP 的语法规则，二是熟悉 Dreamweaver 开发工具的使用。运行结果如图 1.111 所示。

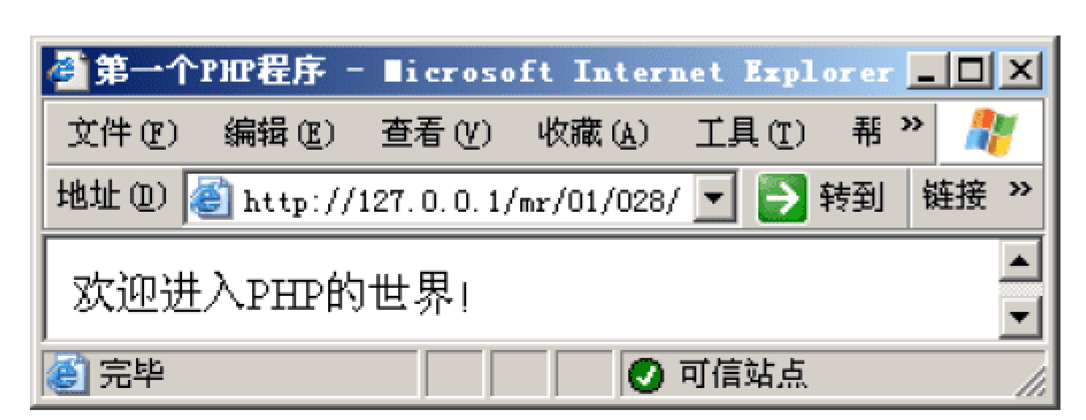


图 1.111　第 1 个 PHP 程序

关键技术

必须将通过 Dreamweaver 开发工具编写的 PHP 脚本，存储到 PHP 开发环境指定的服务器文件夹下才可以运行。

设计过程

（1）打开 Dreamweaver 开发工具，新建一个 PHP 项目，如图 1.112 所示。

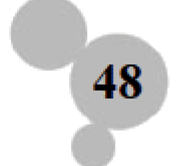




图 1.112　新建一个 PHP 项目

（2）单击图 1.112 中的 PHP 项目图标，打开一个 PHP 项目文件夹，如图 1.113 所示。

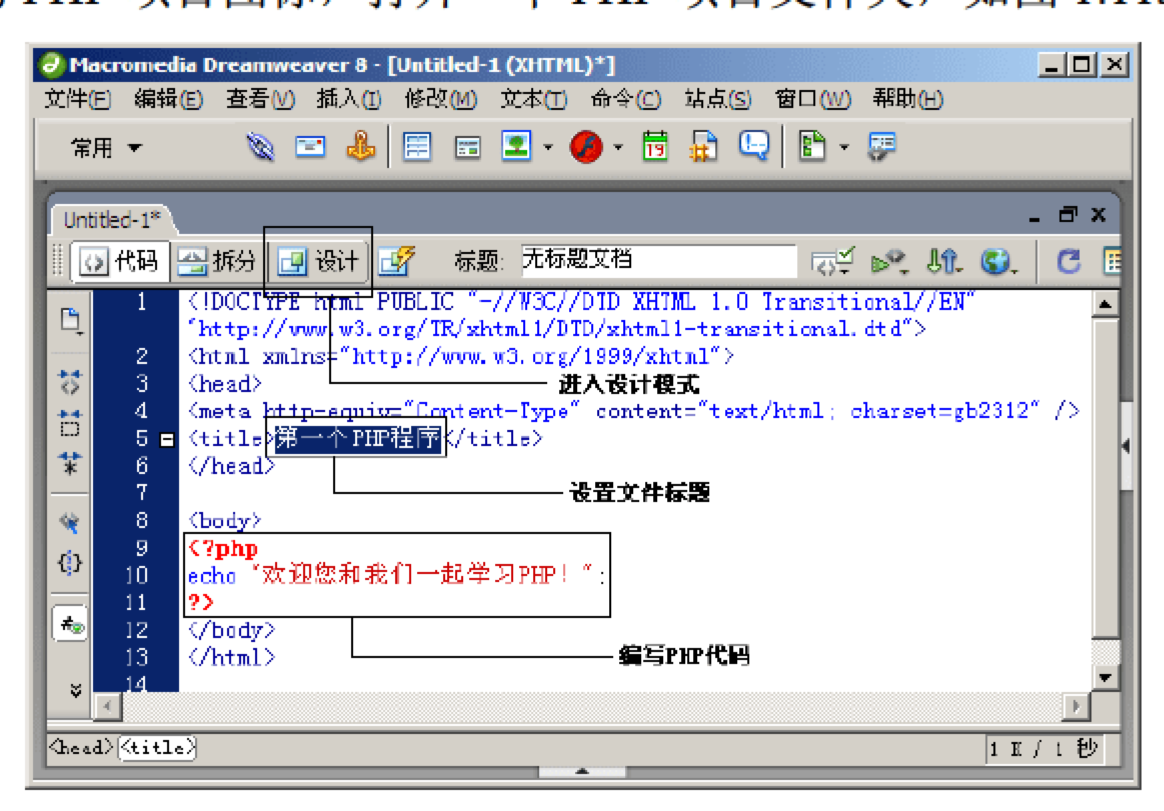


图 1.113　新的 PHP 项目文件

（3）在图 1.113 所示的文件中，首先定义文件的标题，<title>第一个 PHP 程序</title>，然后编写 PHP 代码，其代码如下：

```
<?php
echo "欢迎进入 PHP 的世界！";
?>
```

PHP 代码分析：

☑　“<?php”和“?>”是 PHP 的标记对。在这对标记中的所有代码都被当作 PHP 代码来处理。

☑　echo 是 PHP 中的输出语句，与 ASP 中的 response.write、JSP 中的 out.print 含义相同，输出字符串或者变量值。每行代码都以分号“;”结尾。

（4）保存文件，选择图 1.113 中的“文件”/“另存为”命令，将编写的文件保存在站点的根目录 www 文件夹中，并命名为 test.php，最后单击“保存”按钮，如图 1.114 所示。

■ 秘笈心法

心法领悟 028：运行程序的两种方法。

程序的运行有两种方法：第一种，直接在 Dreamweaver 开发工具中按 F12 键运行程序；第二种，在 IE 浏览器的地址栏中输入 http://127.0.0.1/test.php。程序运行的前提是必须将程序存储到服务器指定的文件夹中。

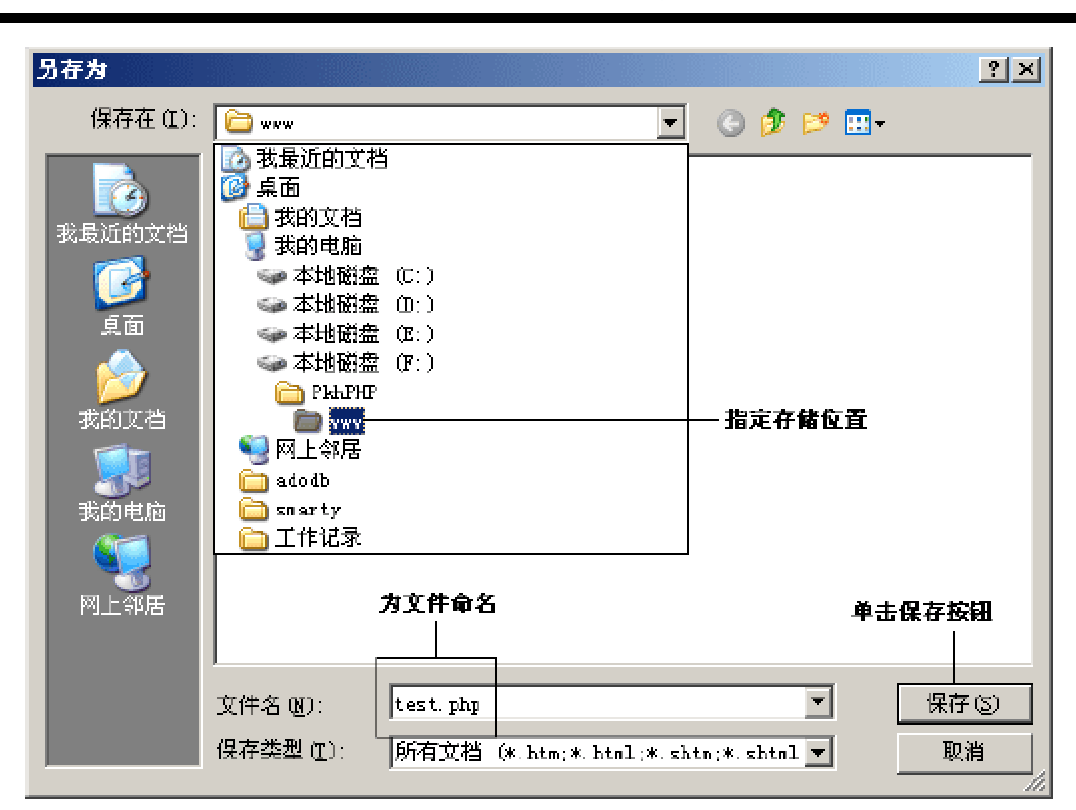
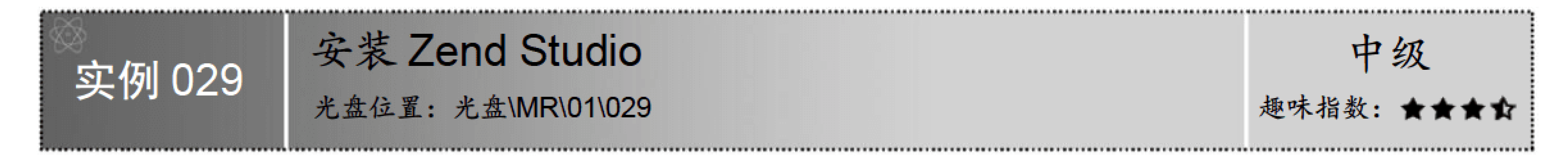


图 1.114　保存编写的 PHP 文件

1.7　Zend Studio 开发工具

Zend Studio 是目前比较流行的 PHP 开发工具，囊括了编辑、调试、配置 PHP 程序所需要的客户及服务器组件，尤其是功能齐全的调试功能，让 PHP 错误不再棘手。

Zend Studio 也有不足之处，一是运行速度慢；二是没有 Dreamweaver 的界面设计功能，只能通过浏览器来浏览页面设计效果；三是其是一款收费软件（但可以下载试用版），下载地址为 http://www.zend.com/store/products/zend-studio.php。



实例说明

在应用 Zend Studio for Eclipse 之前，我们有必要介绍一下它的安装步骤，因为只有正确地完成安装，才能够顺利地应用。

关键技术

在安装 Zend Studio for Eclipse 的过程中，关键是安装内容的选择、安装位置的选择和定义 Zend 支持的文件格式，这是在安装过程中必须注意的内容。

设计过程

（1）运行安装文件，进行安装准备工作，如图 1.115 所示。

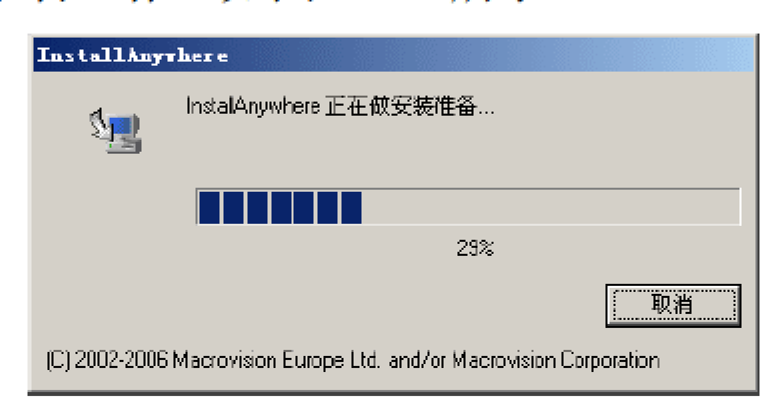
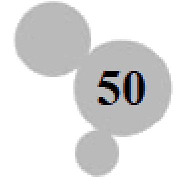
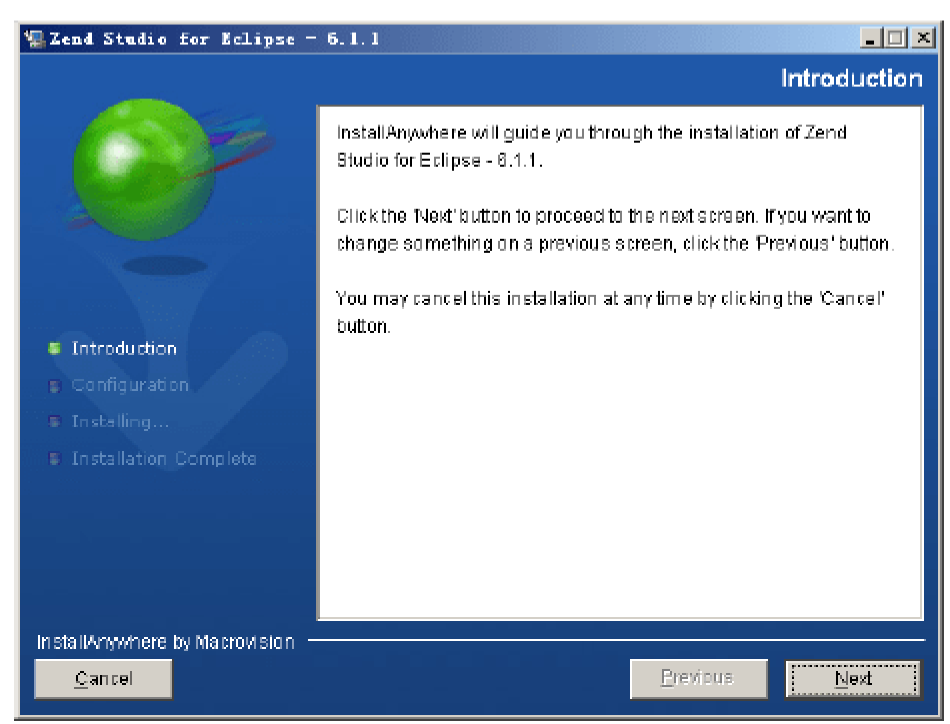


图 1.115　安装 Zend Studio for Eclipse 的准备工作



（2）进入安装界面，如图 1.116 所示，单击 Next 按钮。

（3）进入阅读服务器条款对话框，选择同意，如图 1.117 所示，单击 Next 按钮。



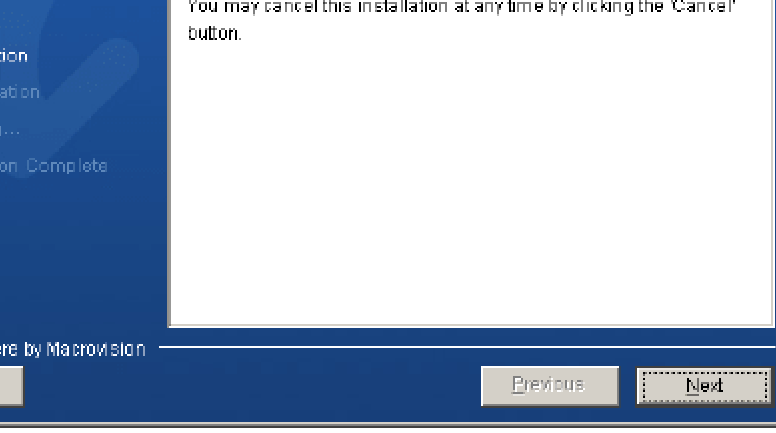
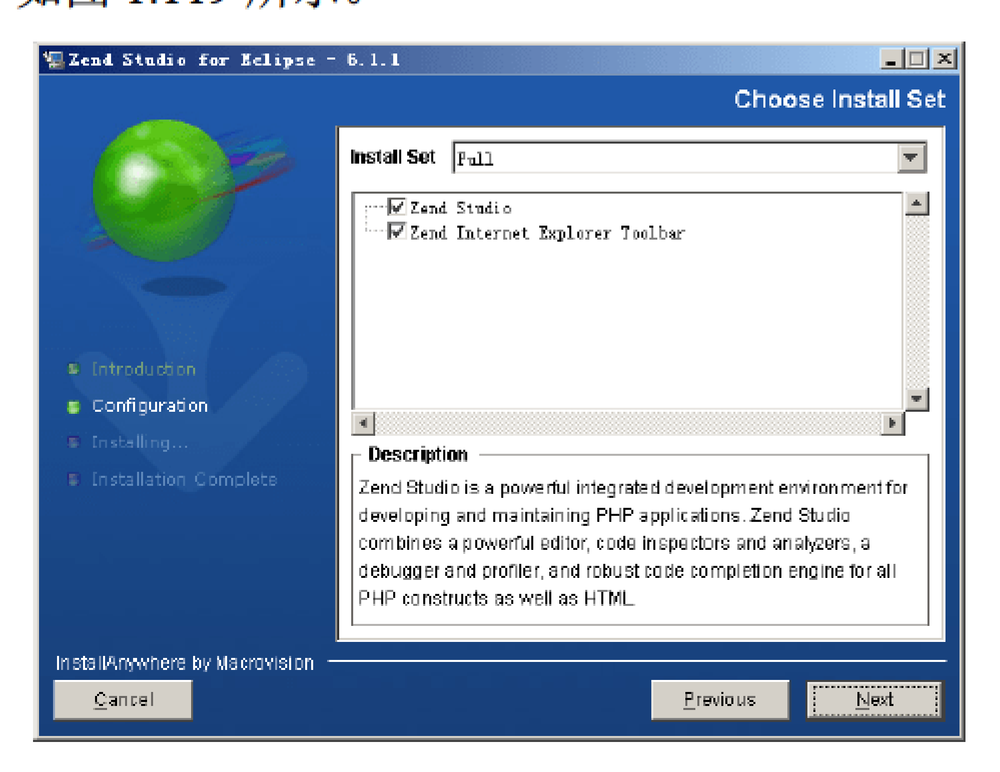


图 1.116　开始进行安装　　　　图 1.117　阅读服务条款

（4）在选择安装内容对话框中包括一个 Zend 浏览器和 Zend Studio，两项全部选中，如图 1.118 所示。单击 Next 按钮，进入下一步。

（5）在进入的对话框中设置文件的安装目录，用户可以任意指定安装位置，确定后单击 Next 按钮进行下一步，如图 1.119 所示。



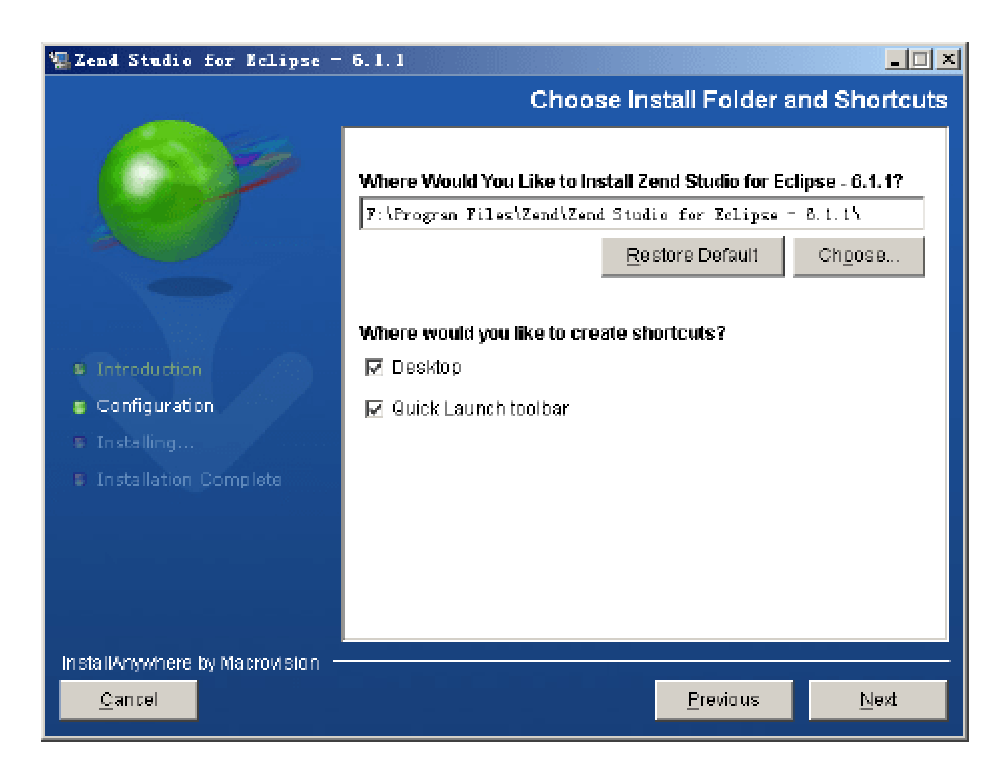
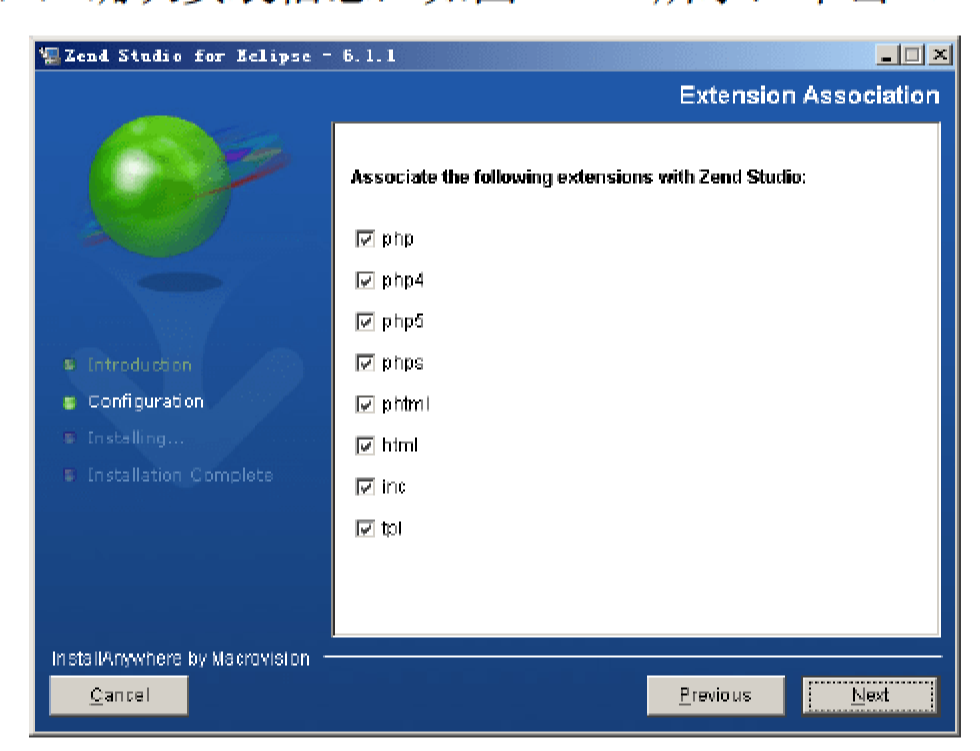



图 1.118　选择安装内容　　　　图 1.119　设置工具安装位置

（6）在进入的定义 Zend 支持的文件格式对话框中，全部选中，如图 1.120 所示。

（7）确认安装信息，如图 1.121 所示，单击 Install 按钮开始进行安装。



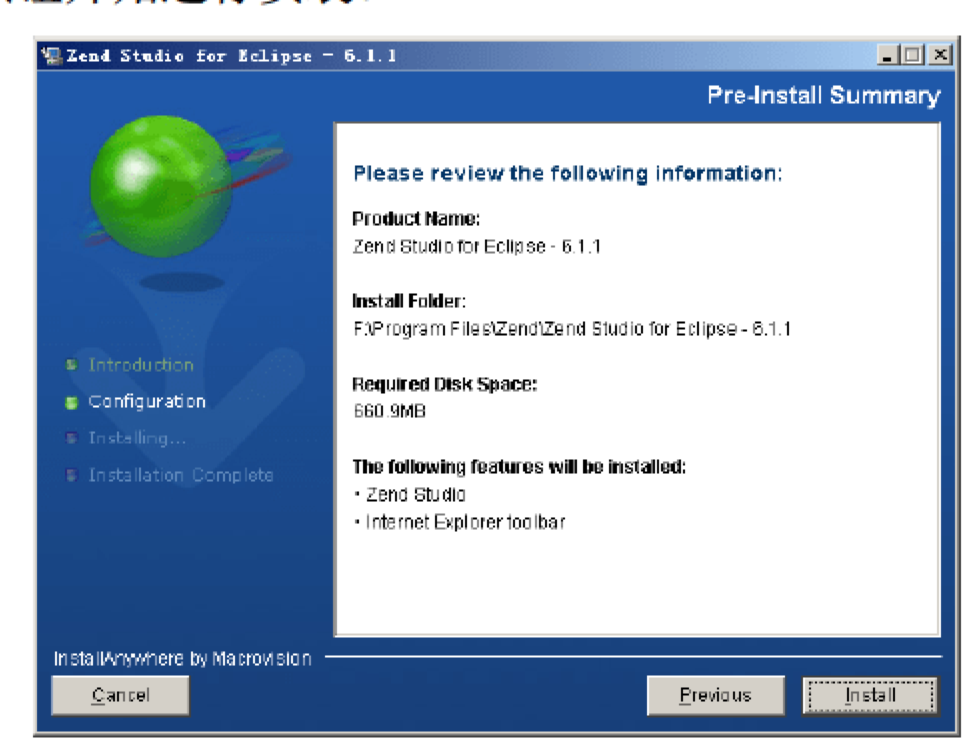


图 1.120　定义 Zend 支持的文件格式　　　　图 1.121　确认安装信息



（8）安装进行中，如图 1.122 所示。

（9）安装完成，单击 Done 按钮，如图 1.123 所示。



图 1.122　Zend 安装进行中

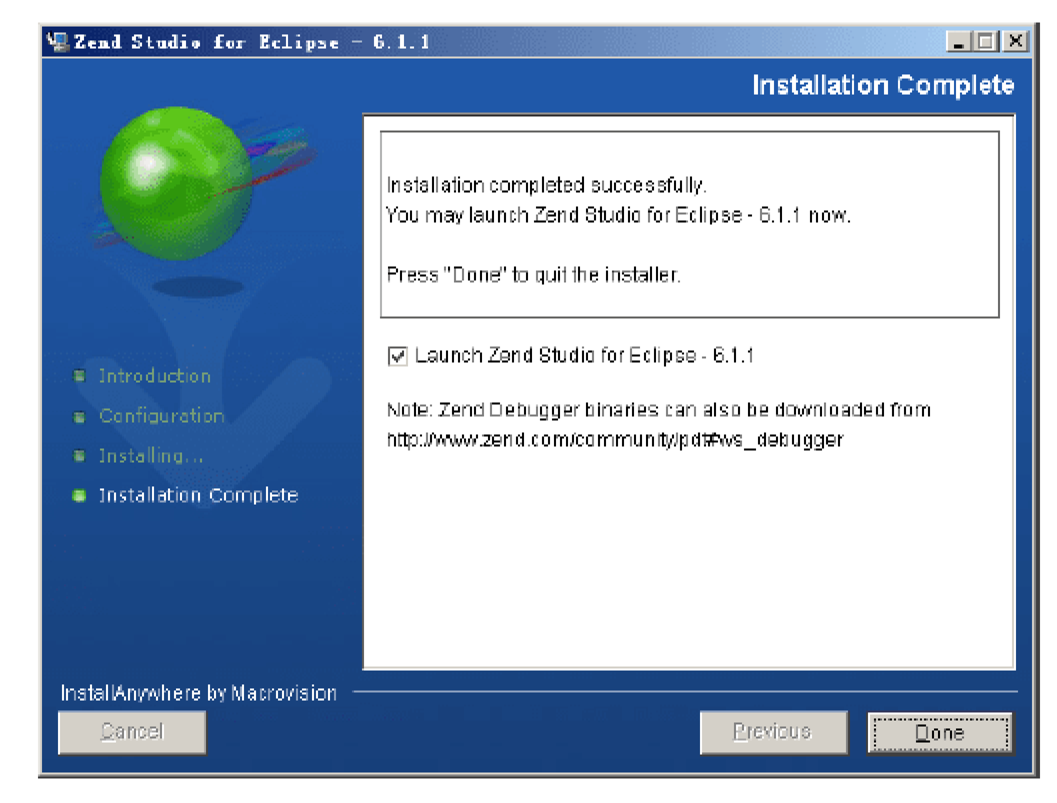


图 1.123　安装成功

这就是 Zend Studio for Eclipse 的安装，过程非常简单，但是等待的时间较长。

秘笈心法

心法领悟 029：选择工作空间。

运行 Zend Studio for Eclipse，可以选择 File/Switch_Workspace 命令来选择工作的空间，或者重新创建工作空间，如图 1.124 所示。

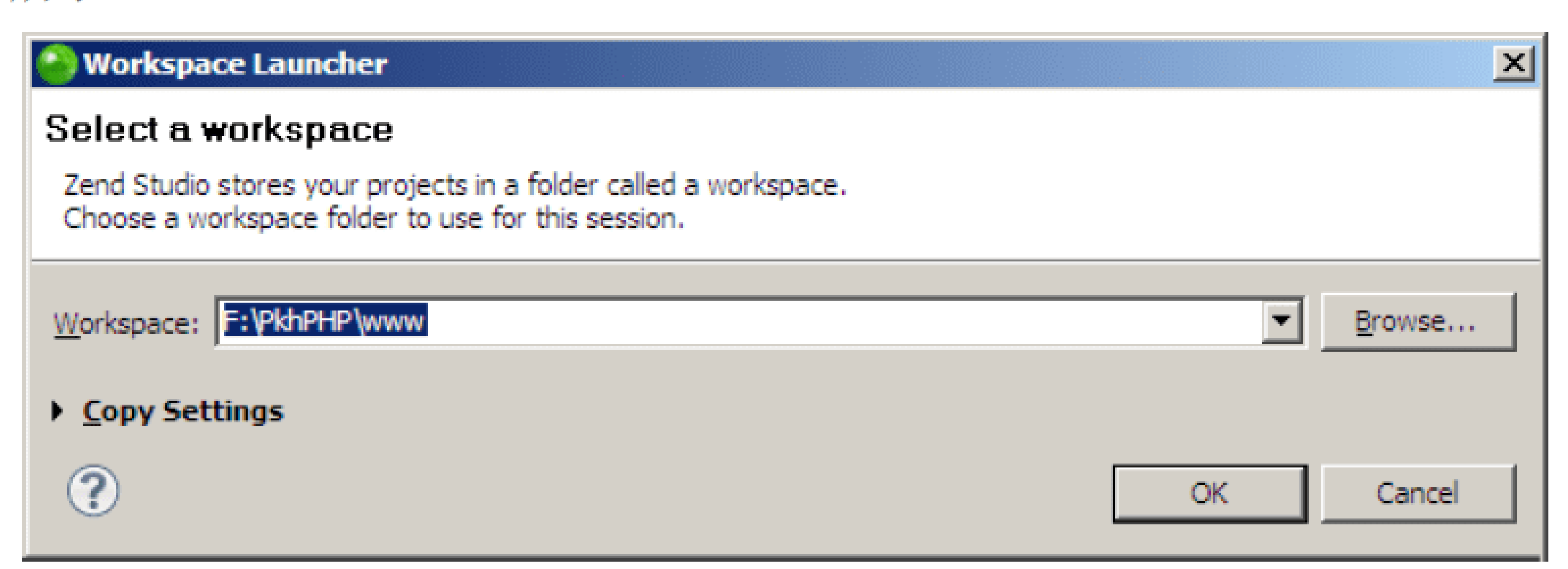


图 1.124　创建工作空间

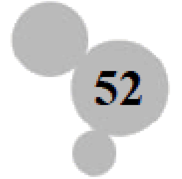
实例 030	Zend Studio 创建 PHP 项目 光盘位置：光盘\MR\01\030	中级 趣味指数：★★★★☆

实例说明

本实例介绍如何通过 Zend Studio 创建 PHP 项目。在本项目中创建一个 PHP 文件，通过 phpinfo()函数输出 PHP 的配置信息，其运行结果如图 1.125 所示。

关键技术

通过 Zend Studio 编写 PHP 程序，首先要选择正确的工作空间，如果工作空间选择不正确，那么 PHP 的程序就无法运行。



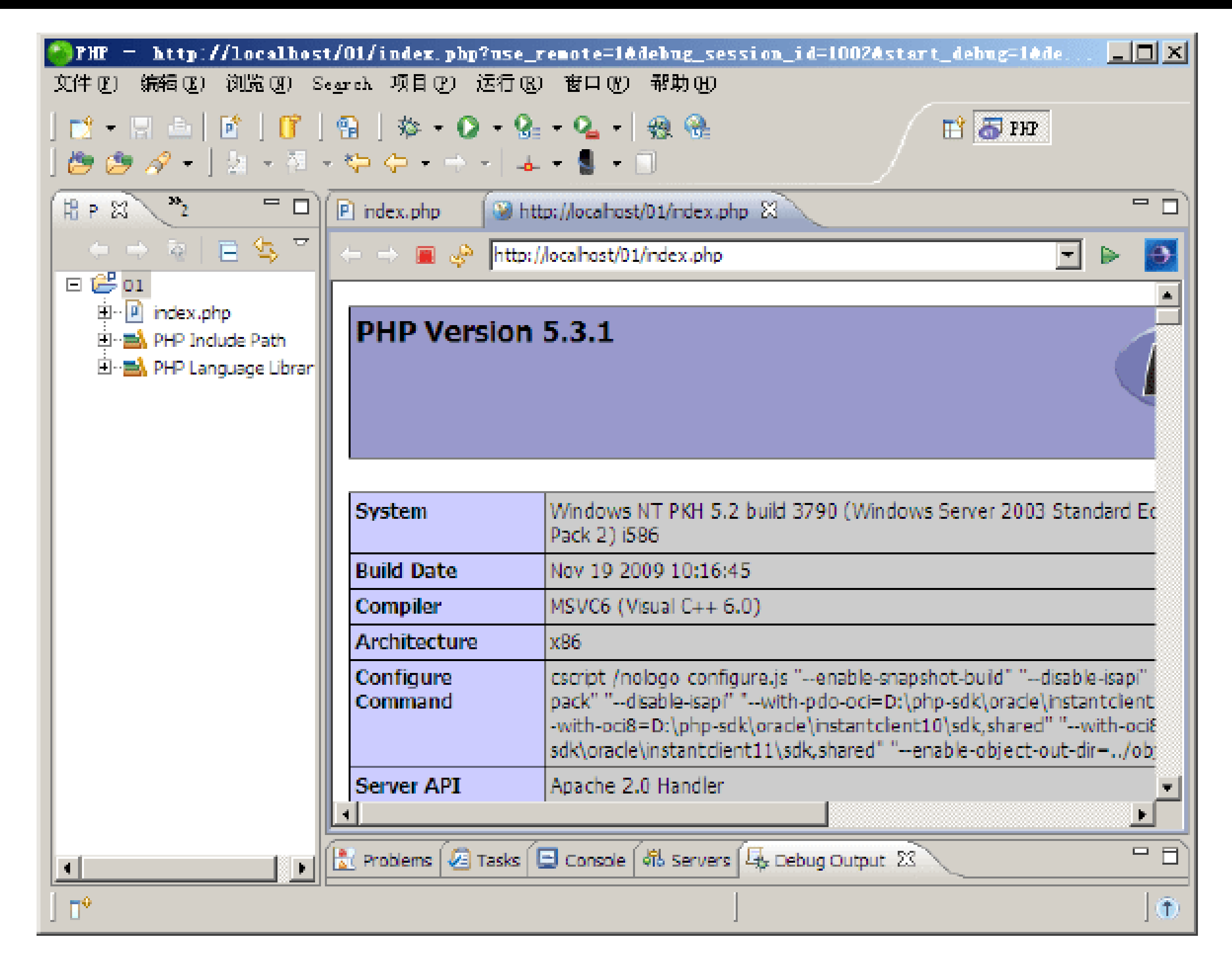


图 1.125　运行程序

设计过程

（1）运行 Zend Studio，首先进入 Zend Studio 的欢迎页面，如图 1.126 所示。

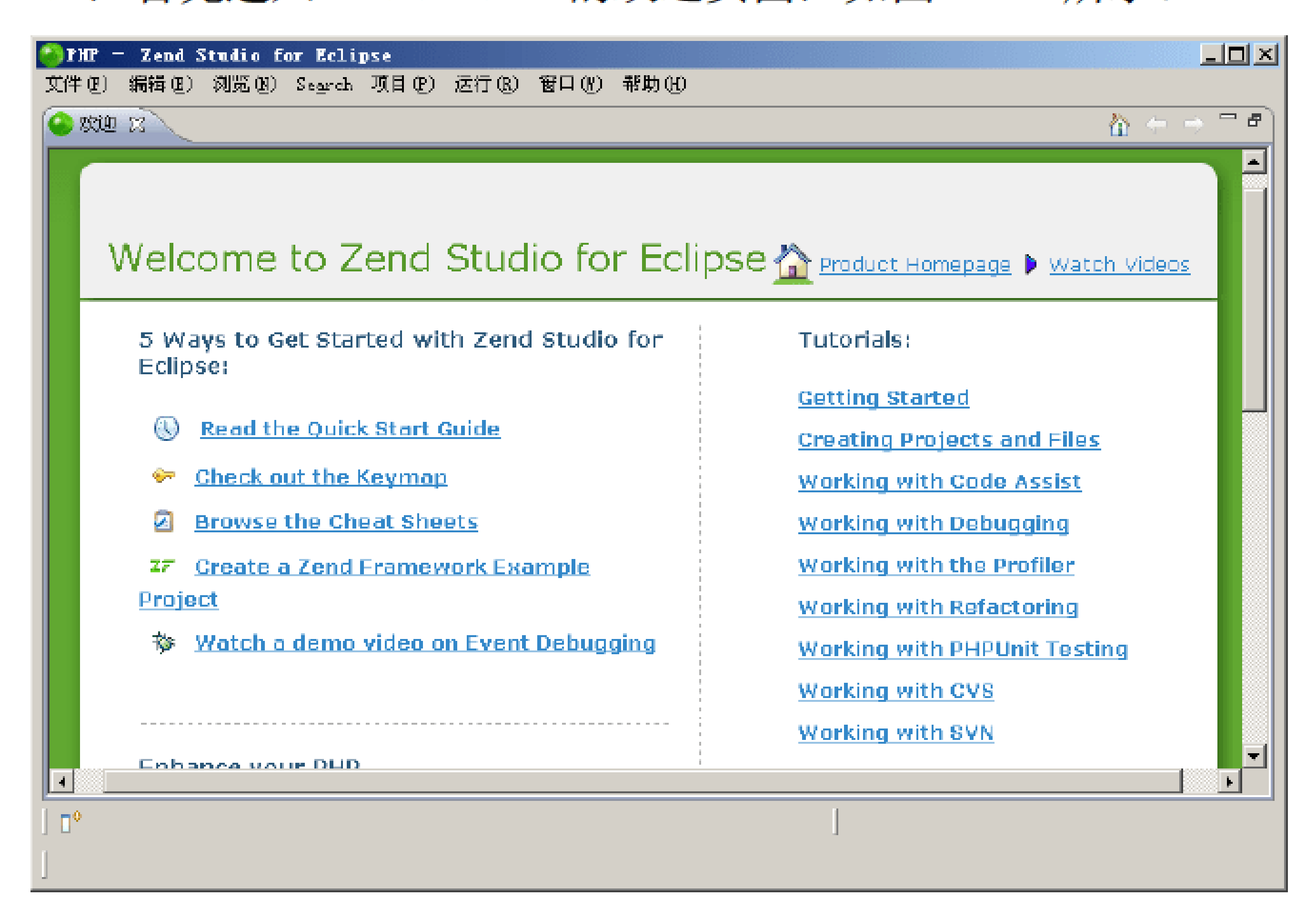


图 1.126　进入 Zend Studio 欢迎页面

（2）选择“文件”/“切换工作空间”/“其他”命令，设置工作空间，将项目保存在指定名称的文件夹中，如图 1.127 所示，将 F:\PkhPHP\www 文件夹作为工作空间。

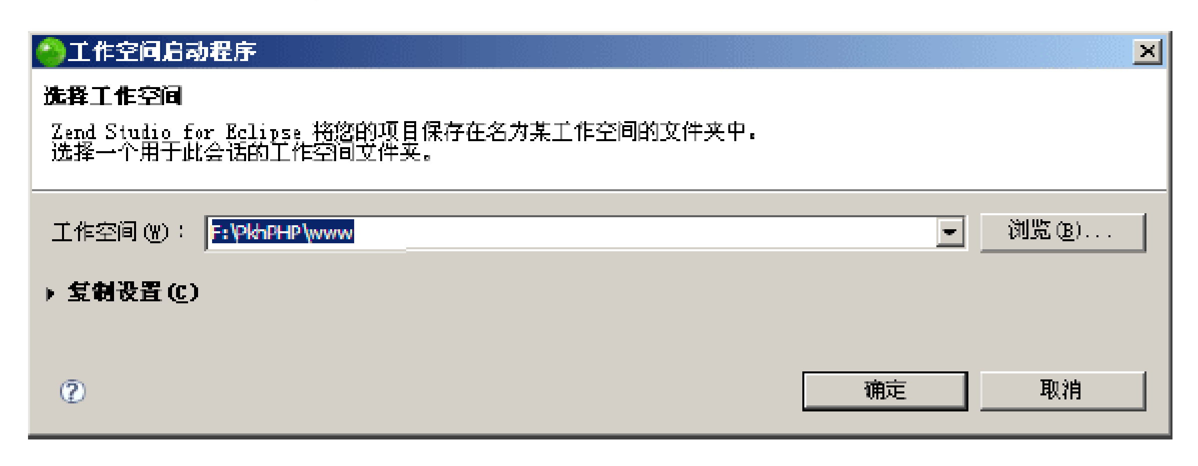


图 1.127　选择工作空间

（3）接下来进入 Zend Studio 的工作台中，如图 1.128 所示。

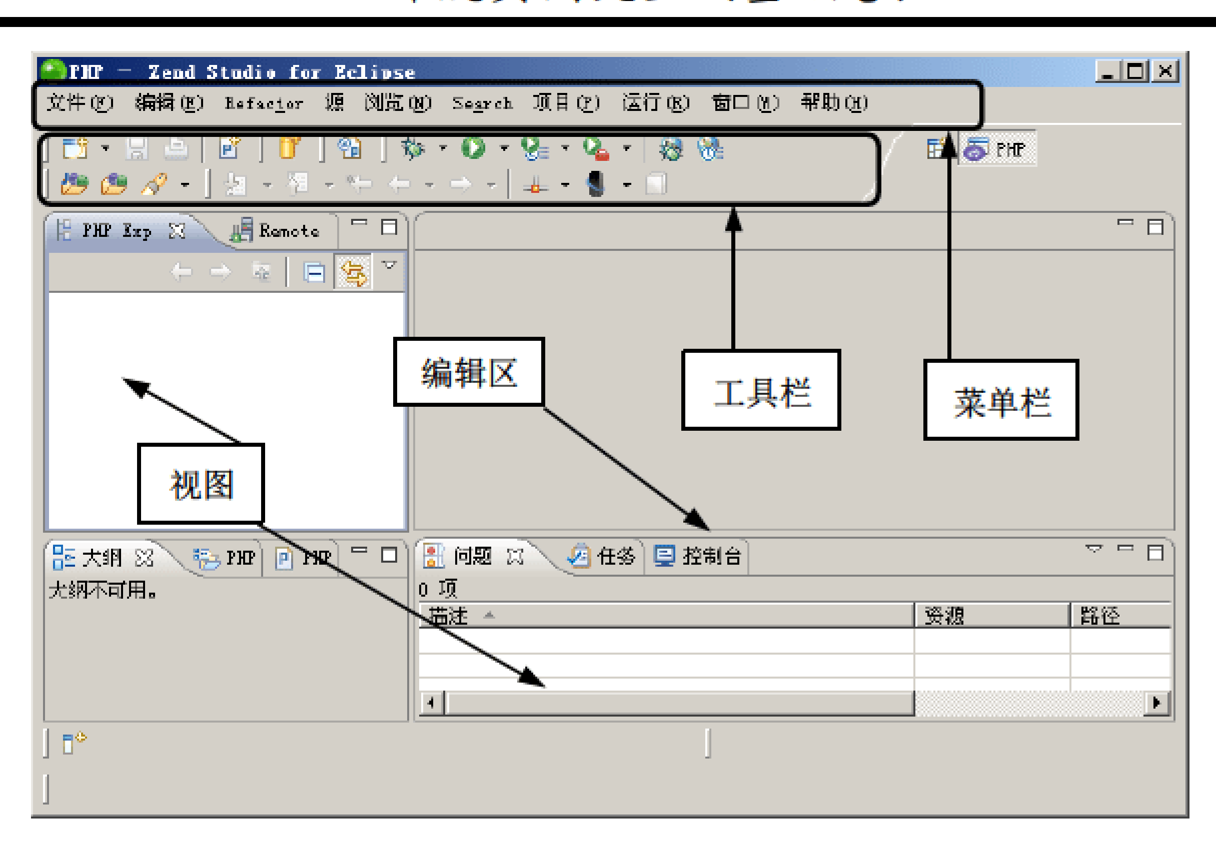


图 1.128　Zend Studio 的工作台

（4）创建一个 PHP 项目。选择“文件”/“新建”/PHP Project 命令，创建一个 PHP 项目，如图 1.129 所示。

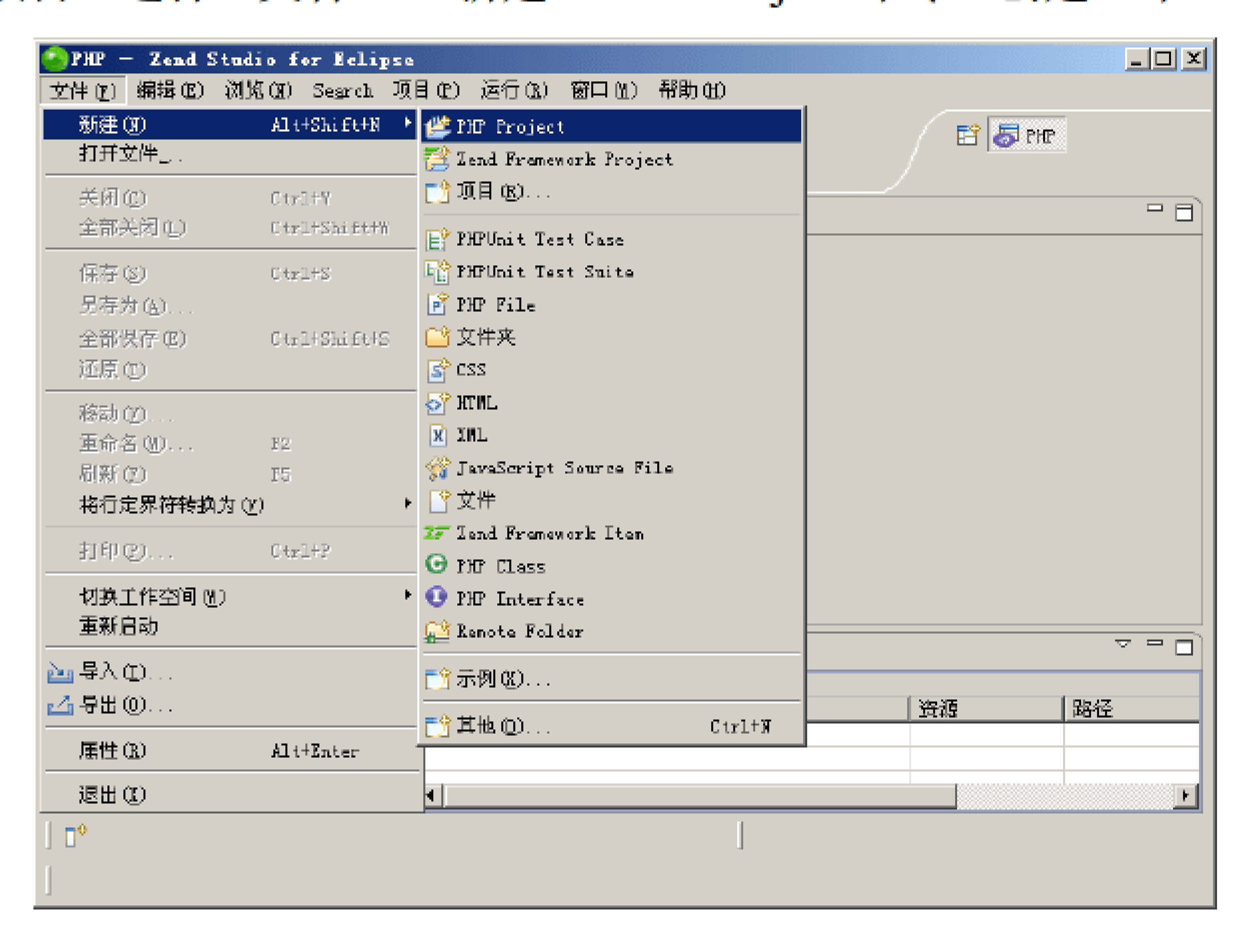


图 1.129　创建 PHP 项目

（5）进入新项目创建窗口中，在该窗口中设置项目名称为 01；指定项目文件存储位置，这里使用默认值；选择 PHP 的版本，设置是否支持 JavaScript 脚本，如图 1.130 所示。

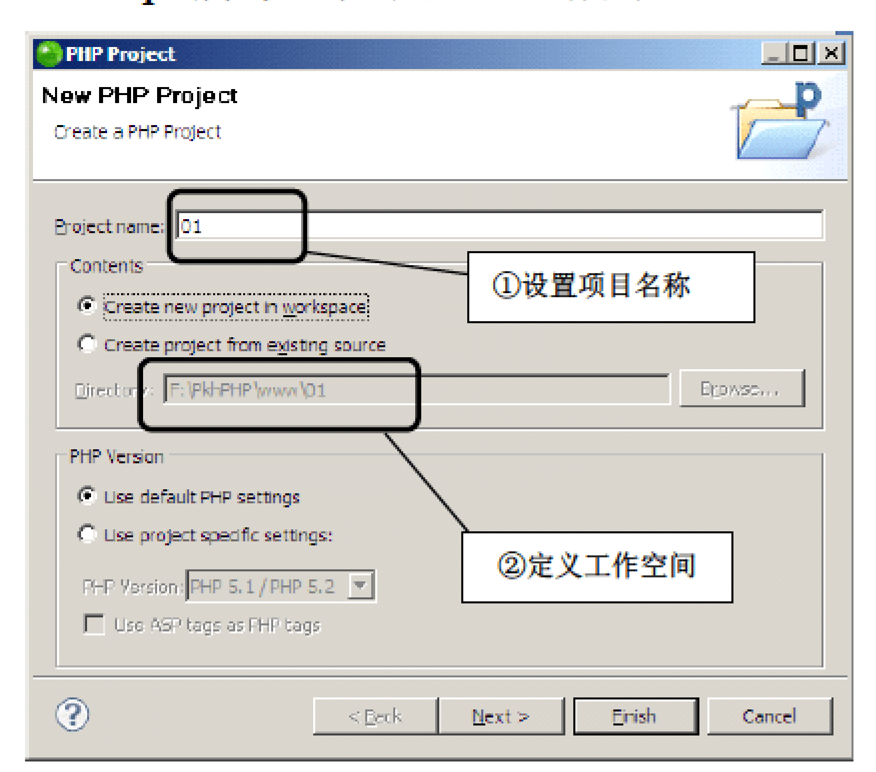
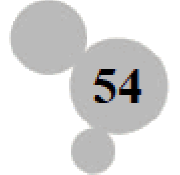


图 1.130　新建项目

（6）在项目 01 中创建 PHP 文件，如图 1.131 所示。



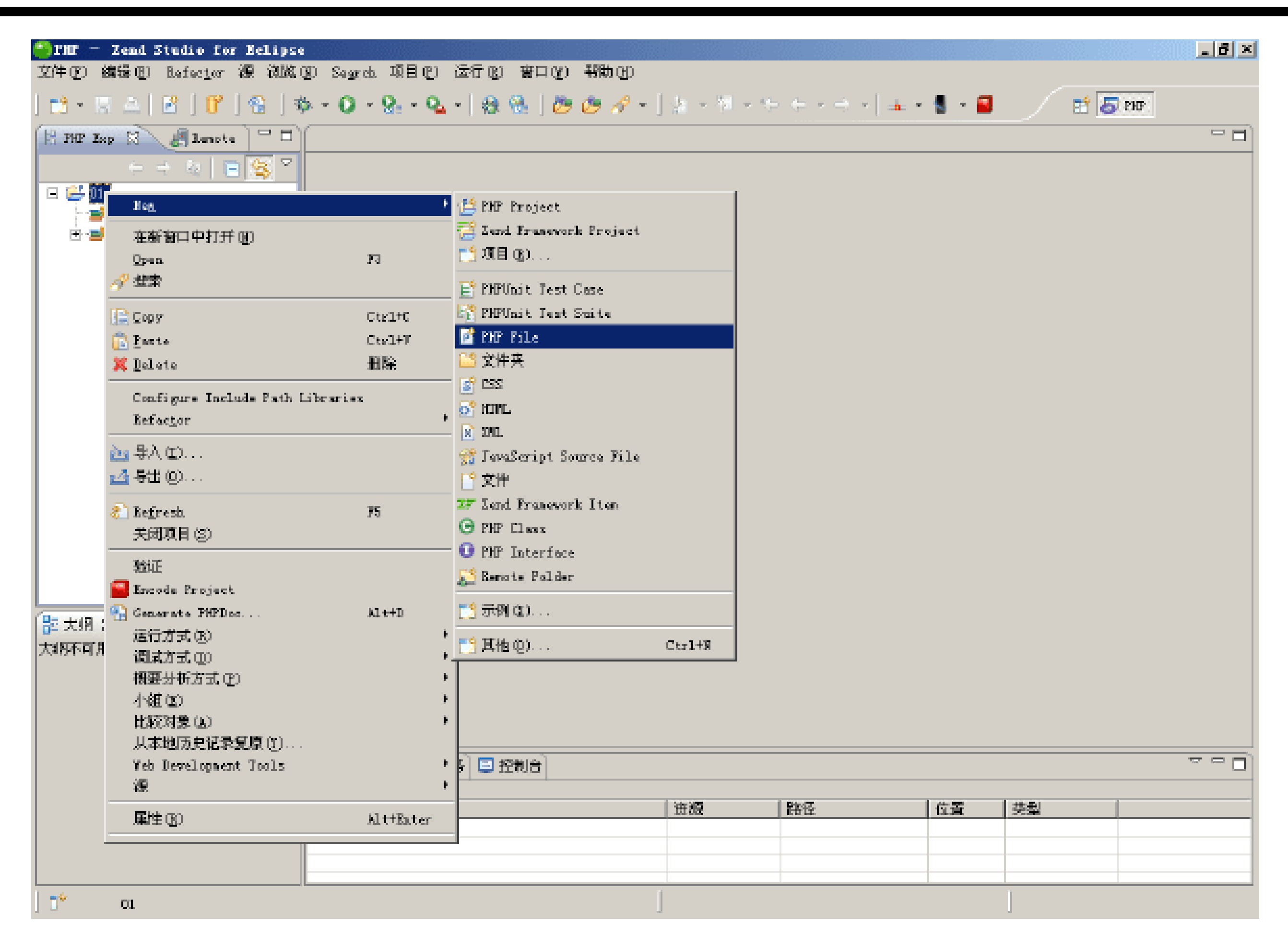


图 1.131　创建 PHP 文件

在图 1.129 所示的操作步骤中，不但可以创建 PHP 文件，还可以创建 HTML、CSS、JavaScript 和 XML 文件等，具体可以根据程序的实际情况进行创建。

（7）进入文件创建窗口，定义文件名称，最后单击 Finish 按钮，如图 1.132 所示。

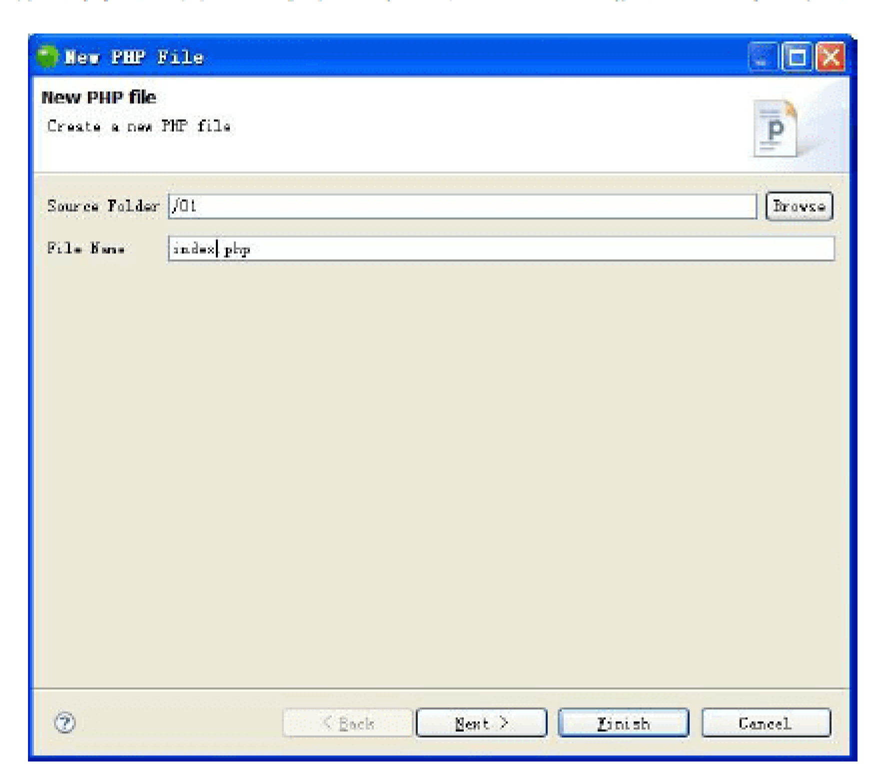


图 1.132　创建 index.php 文件

（8）文件创建成功后，就可以在编辑区中编辑文件的内容，如图 1.133 所示。

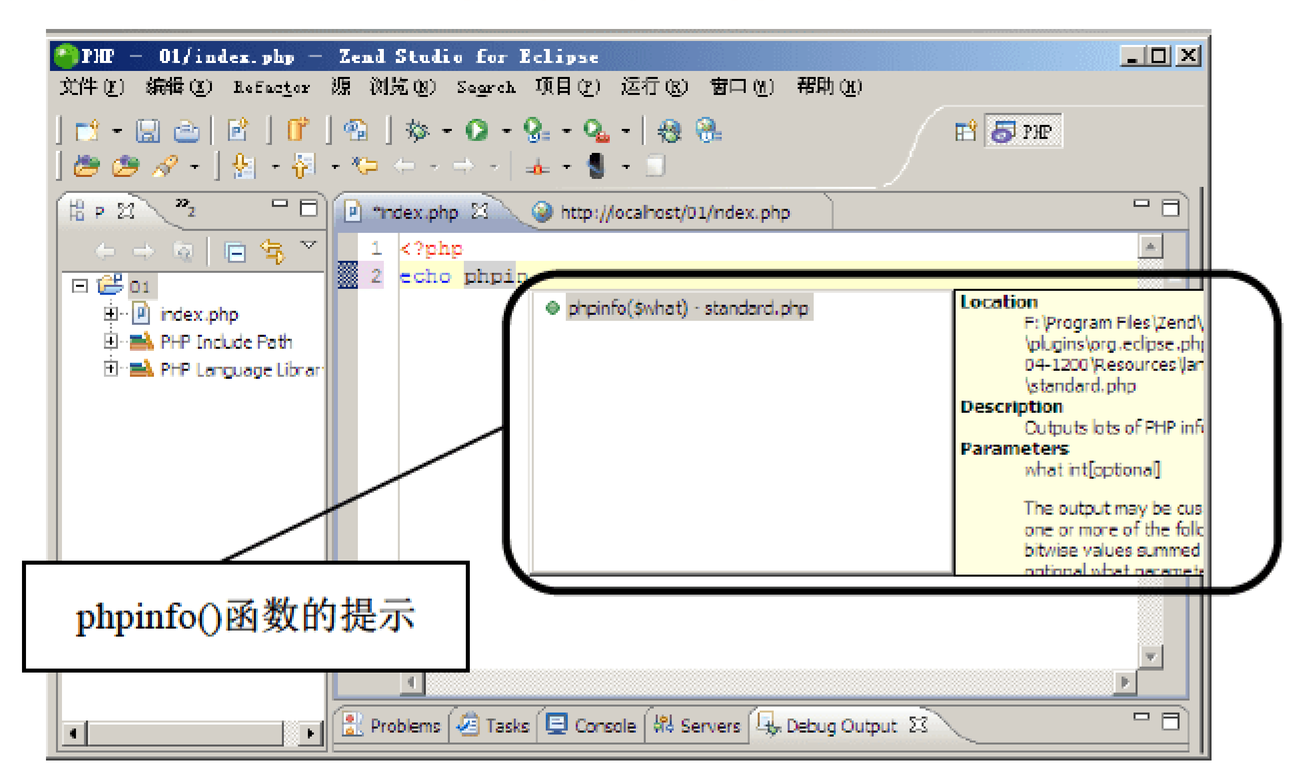


图 1.133　编辑 PHP 代码

如图 1.133 所示，Zend Studio for Eclipse 工具最大的优点就是代码提示功能，它可以在用户输入部分 PHP 脚本语言时，给出相关 PHP 脚本语言的提示供选择。这里应用 phpinfo()函数获取 PHP 的相关配置信息。

（9）编辑完成后，保存该文件，单击按钮，运行程序，运行结果请参看实例说明。

秘笈心法

心法领悟 030：Zend Studio 对 Zend Framework 框架的支持。

通过 Zend Studio 不但可以创建 PHP 项目，而且可以直接生成 Zend Framework 框架，开发者不必再为搭建 Zend Framework 框架而头疼。

实例 031 Zend Studio 编码格式的转换

光盘位置：光盘\MR\01\031

中级

趣味指数：★★★★☆

实例说明

在 Zend Studio 中同样可以控制创建项目、文件的编码格式，至于编码格式选择的重要性这里不再赘述。下面介绍如何在 Zend Studio 中设置项目、文件的编码格式。如果通过 Zend Studio 打开一个编码格式不匹配的文件，输出结果如图 1.134 所示。

newfile.php　index_ok.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
<title>Dreamweaver鎺у埗娴忚　錯儿播寮哈楀彛</title>
<link href="css/styles.css" rel="stylesheet" type="text/css" />
<script type="text/JavaScript">
<!--
function MM_popupMsg(msg) { //v1.0
  alert(msg);
}
//-->
</script>
</head>
```

图 1.134　编码格式不正确导致乱码

关键技术

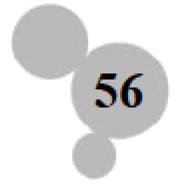
在 Zend Studio 中设置项目、文件的编码格式，关键是从根本着手，也就是说要控制一个项目中文件的编码格式，只要项目本身的编码格式确立，那么创建的文件格式自然是与项目相同的。

设计过程

（1）在 Zend Studio 中要设置项目的编码格式，首先就要创建一个项目。有关项目创建的具体步骤请参看实例 030，这里不再赘述。

（2）项目创建成功后，选中创建的项目，右击，在弹出的快捷菜单中选择 Properties 命令，弹出如图 1.135 所示的窗口。

（3）在该窗口中选中 Other 单选按钮，在其下拉列表中选择指定的编码格式，最后单击 OK 按钮，项目编码格式即设置完成。



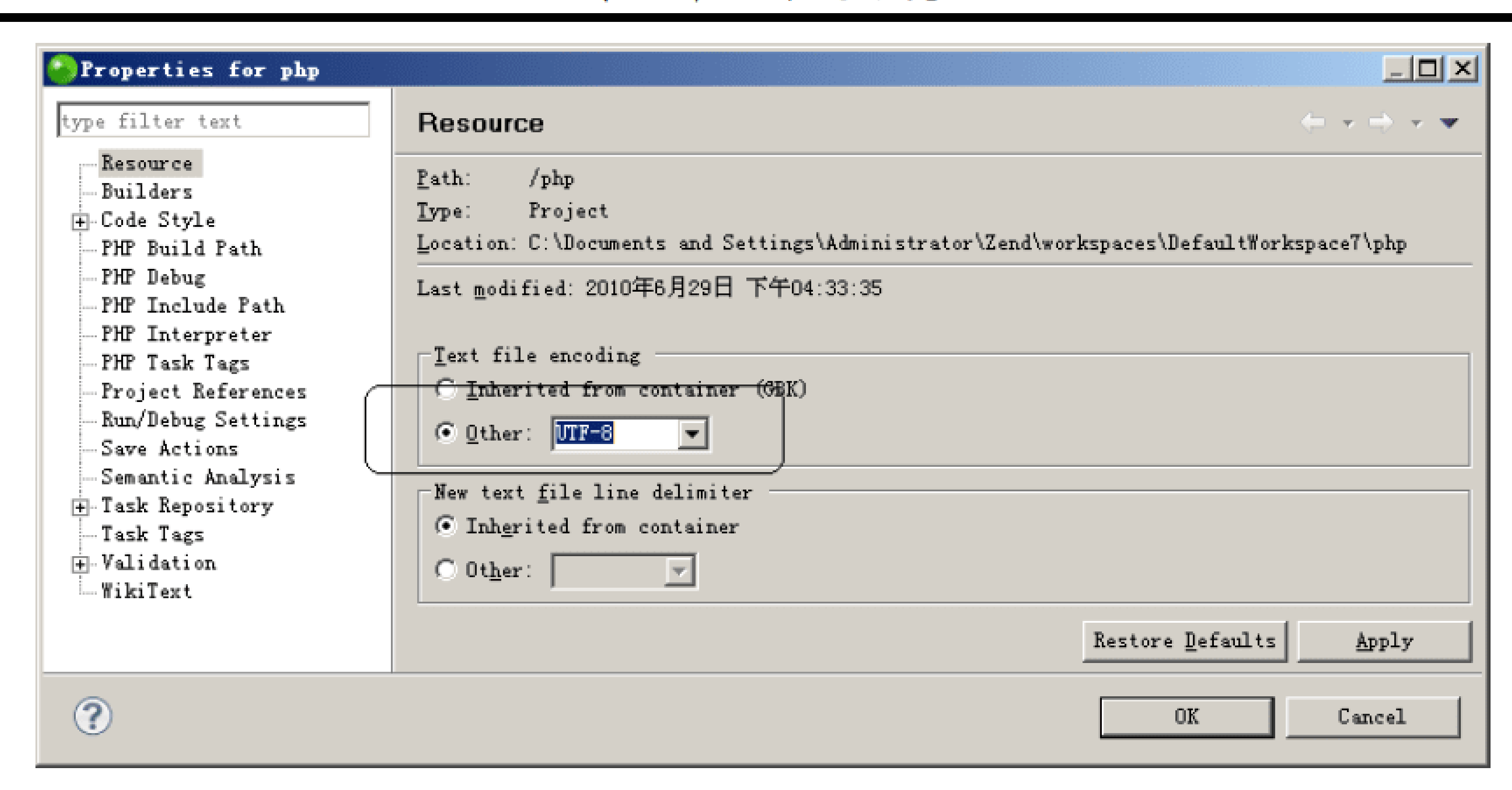


图 1.135　设置项目的编码格式

秘笈心法

心法领悟 031：在 Zend Studio 中修改编码格式。

在实例说明中我们曾看到一个编码格式不匹配的文件打开的效果，那么如何对这个文件的编码格式进行修改呢？

首先，打开编码格式不匹配的文件。然后单击 Edit 按钮，在弹出的菜单中选择 Set Encoding 命令，将弹出如图 1.136 所示的对话框，在这个对话框中选中 Other 单选按钮，设置这个文件的编码格式。最后单击 OK 按钮。

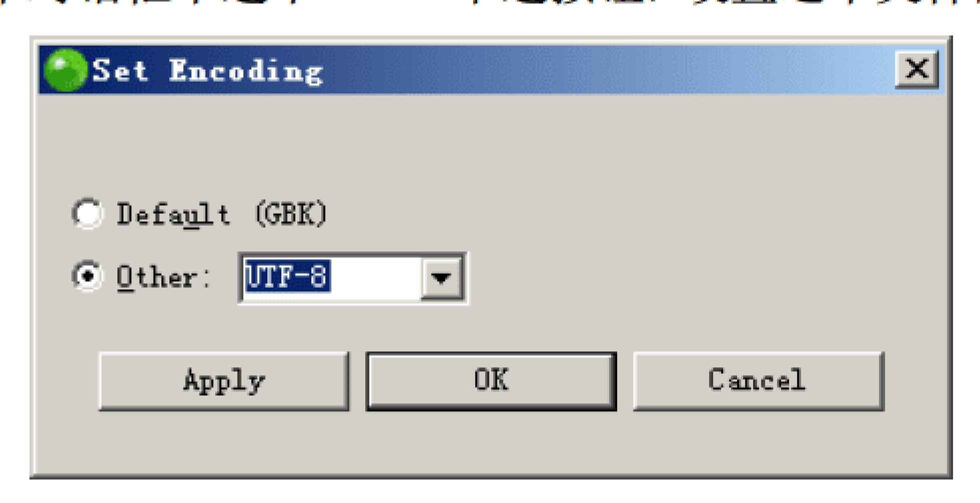


图 1.136　设置文件的编码格式

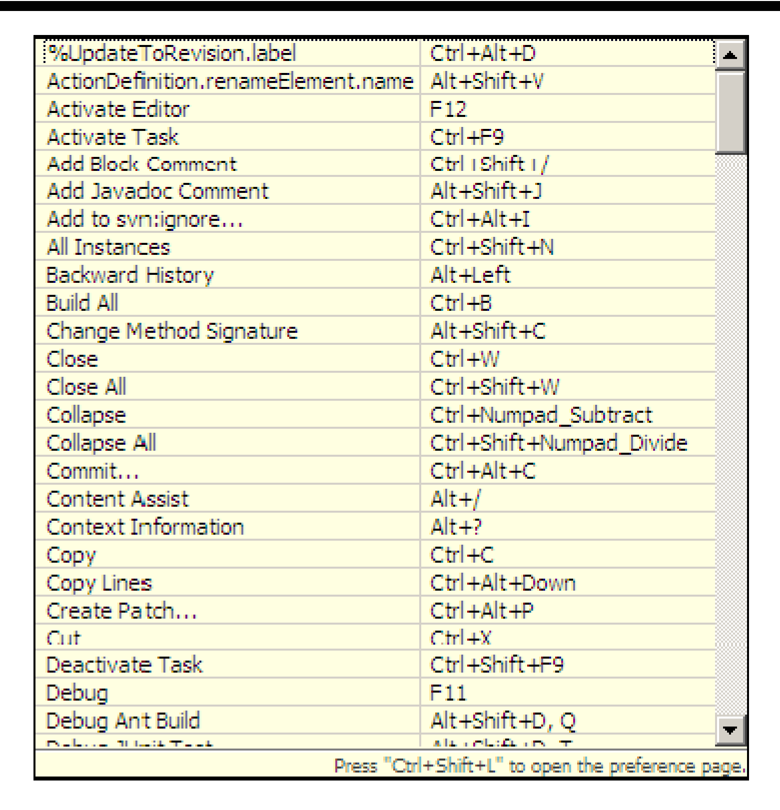
实例 032 | Zend Studio 中快捷键的运用 | 中级
光盘位置：光盘\MR\01\032 | 趣味指数：★★★☆

实例说明

在 Zend Studio 中，通过快捷键可以完成很多操作，快捷键的运用可以为开发者节省很多时间，而且可以简化操作的流程。在本实例中将介绍如何运用 Zend Studio 中的快捷键创建 PHP 项目。

关键技术

要应用 Zend Studio 中的快捷键，必须了解快捷键代表的意义。在 Zend Studio 中，选择“帮助”/Key Assist 命令将弹出 Zend Studio 中的快捷键列表，或者也可以直接按 Ctrl+Shift+L 键，同样可以弹出快捷键列表。在这个列表中可以查看到所有快捷键代表的意义，如图 1.137 所示。



%UpdateToRevision.label	Ctrl+Alt+D
ActionDefinition.renameElement.name	Alt+Shift+V
Activate Editor	F12
Activate Task	Ctrl+F9
Add Block Comment	Ctrl+Shift+/
Add Javadoc Comment	Alt+Shift+J
Add to svn:ignore...	Ctrl+Alt+I
All Instances	Ctrl+Shift+N
Backward History	Alt+Left
Build All	Ctrl+B
Change Method Signature	Alt+Shift+C
Close	Ctrl+W
Close All	Ctrl+Shift+W
Collapse	Ctrl+Numpad_Subtract
Collapse All	Ctrl+Shift+Numpad_Divide
Commit...	Ctrl+Alt+C
Content Assist	Alt+/
Context Information	Alt+?
Copy	Ctrl+C
Copy Lines	Ctrl+Alt+Down
Create Patch...	Ctrl+Alt+P
Cut	Ctrl+X
Deactivate Task	Ctrl+Shift+F9
Debug	F11
Debug Ant Build	Alt+Shift+D, Q

Press "Ctrl+Shift+L" to open the preference page.

图 1.137　Zend Studio 中的快捷键列表

■ 设计过程

（1）在 Zend Studio 中创建一个 PHP 项目。可以直接使用快捷键 Alt+Shift+N，将弹出一个菜单命令列表，选择 PHP Project 命令，创建一个 PHP 项目，如图 1.138 所示。

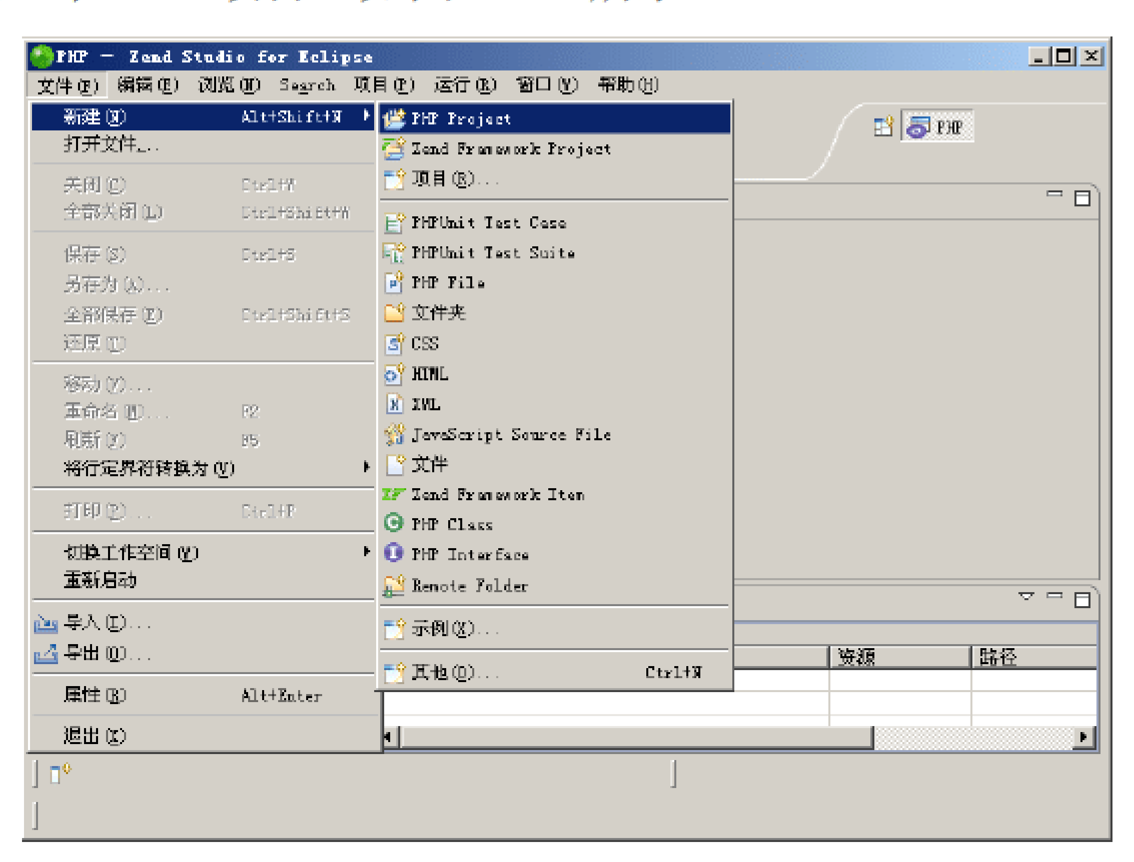


图 1.138　创建 PHP 项目

（2）进入新项目创建窗口中，在该窗口中设置项目名称为 01；指定项目文件存储位置，这里使用默认值；选择 PHP 的版本，设置是否支持 JavaScript 脚本，如图 1.139 所示。

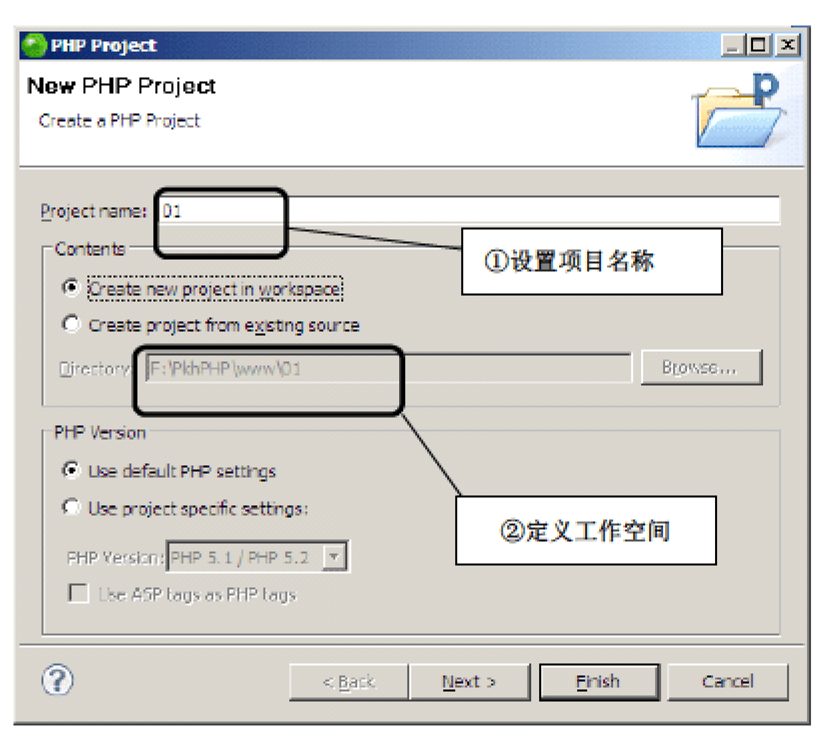
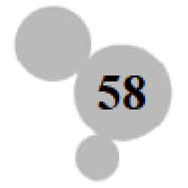


图 1.139　新建项目



（3）在项目 01 中创建 PHP 文件。编写一个简单实例，同样可以使用快捷键 Alt+Shift+N，只是要在列表中选择 PHP File 命令，如图 1.140 所示。

（4）进入文件创建窗口，定义文件名称，最后单击 Finish 按钮，完成文件的创建，如图 1.141 所示。

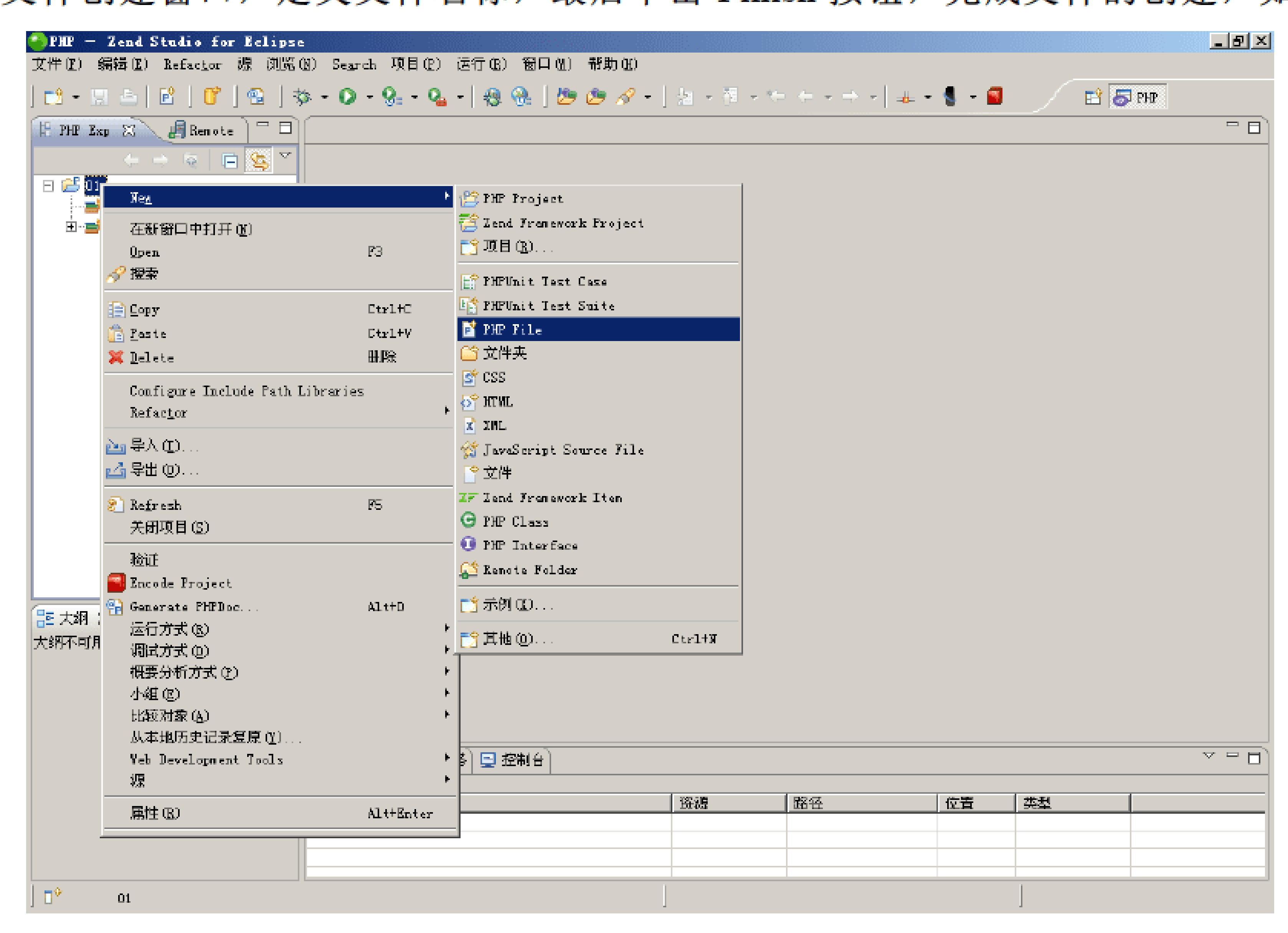


图 1.140　创建 PHP 文件

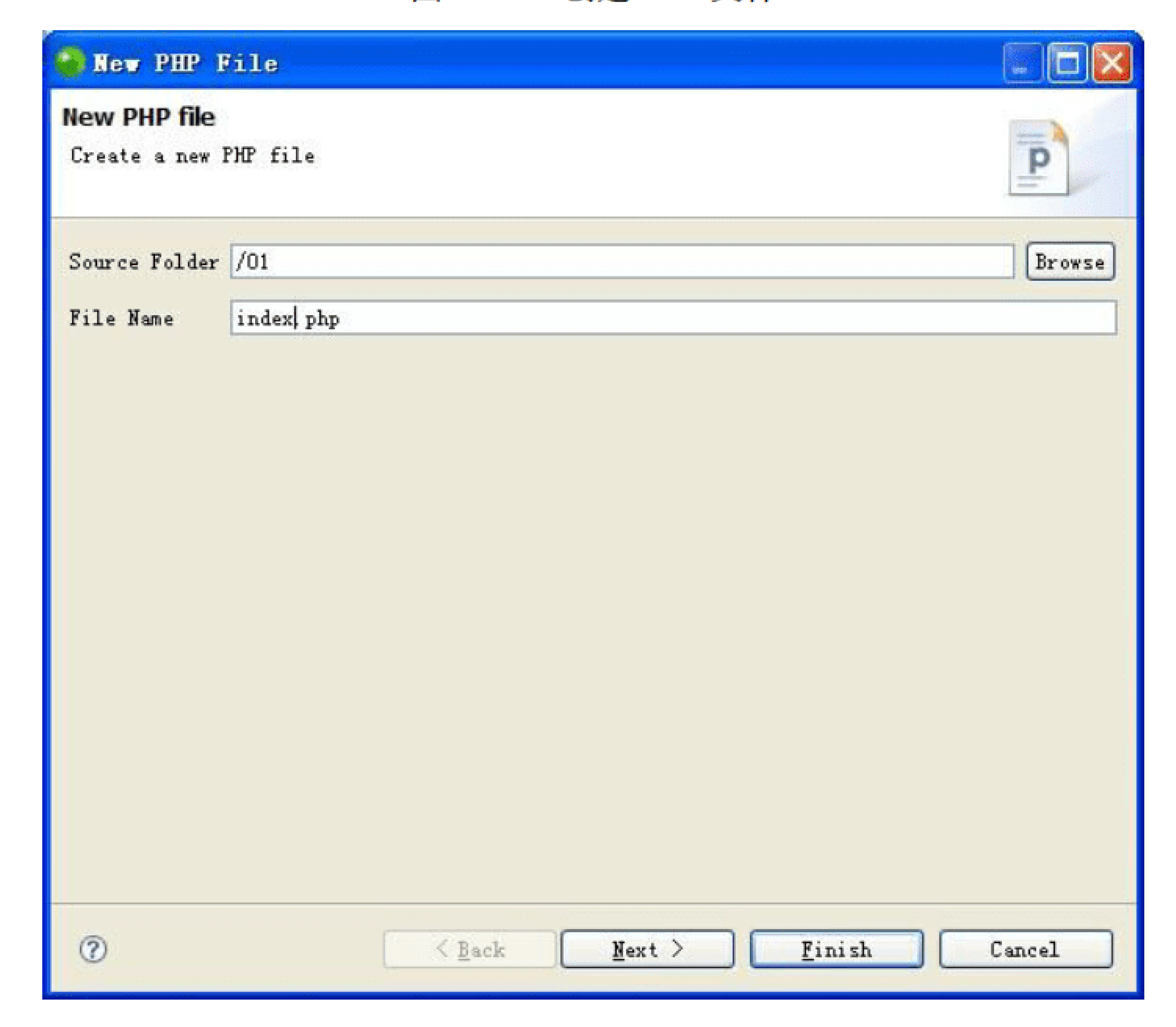


图 1.141　创建 index.php 文件

（5）文件创建成功后，就可以在编辑区中编辑文件的内容。在编辑区中可以通过快捷键 Ctrl+Shift+F 对代码进行格式化，如图 1.142 所示。

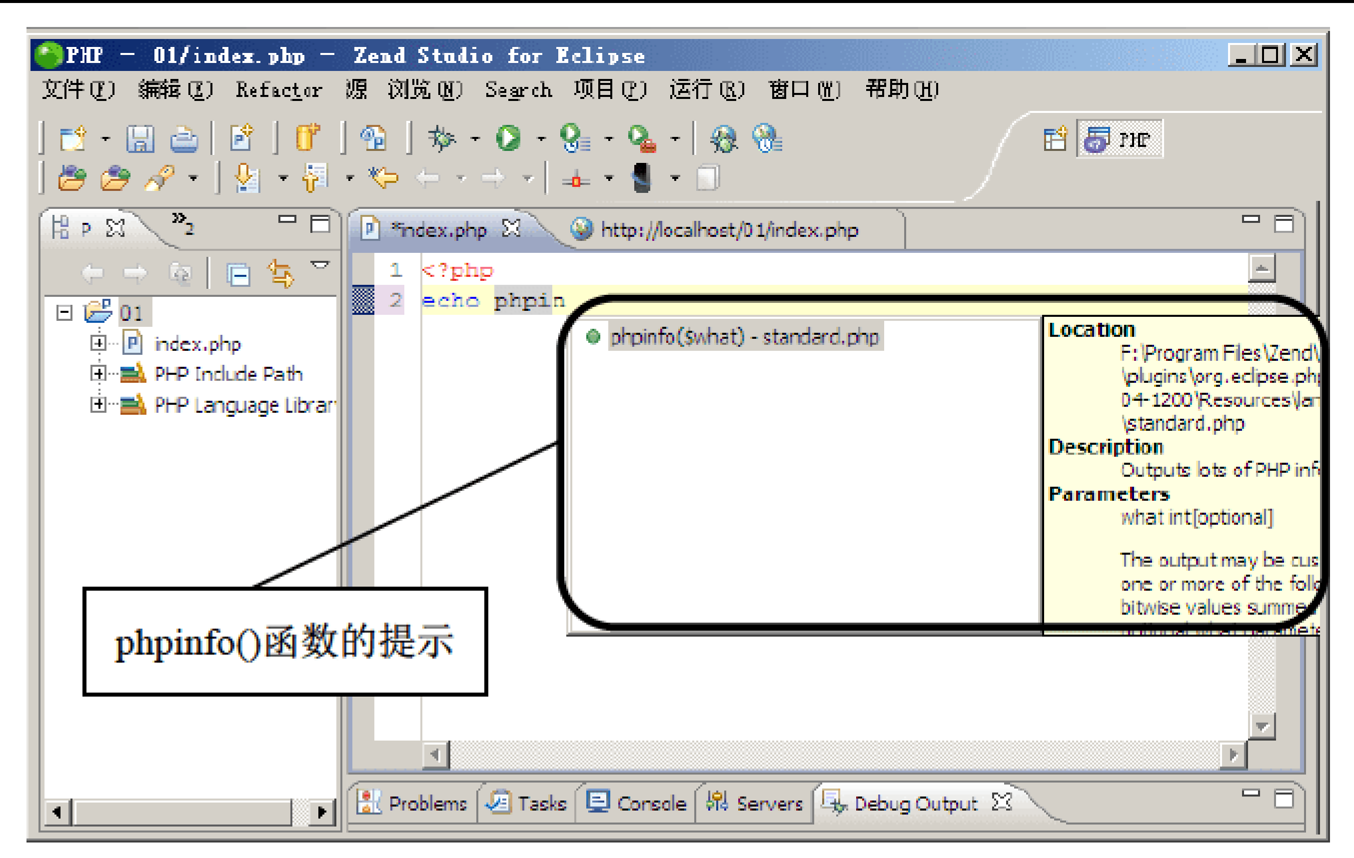


图 1.142　编辑 PHP 代码

（6）编辑完成后，保存该文件，单击按钮，运行程序，运行结果如图 1.125 所示。

秘笈心法

心法领悟 032：在 Zend Studio 中使用快捷键。

在 Zend Studio 中，对于快捷键的使用不必机械地去记忆，只要了解一些常用的快捷键即可。对于初学者来说，通过鼠标进行操作比使用快捷键更好，更有助于对工具功能的理解和学习。

实例 033　Zend Studio 中部署 Apache 服务器

光盘位置：光盘\MR\01\033

中级

趣味指数：★★★☆

实例说明

在运行 PHP 程序的过程中，需要将 PHP 程序复制到 Apache 服务器指定的根目录下，然后在 IE 浏览器的地址栏中输入 http://127.0.0.1/或者 http://localhost/来运行 PHP 程序。而在 Zend Studio 中可以直接部署 Apache 服务器，部署完成后直接单击按钮，运行 PHP 程序，其运行结果如图 1.143 所示。

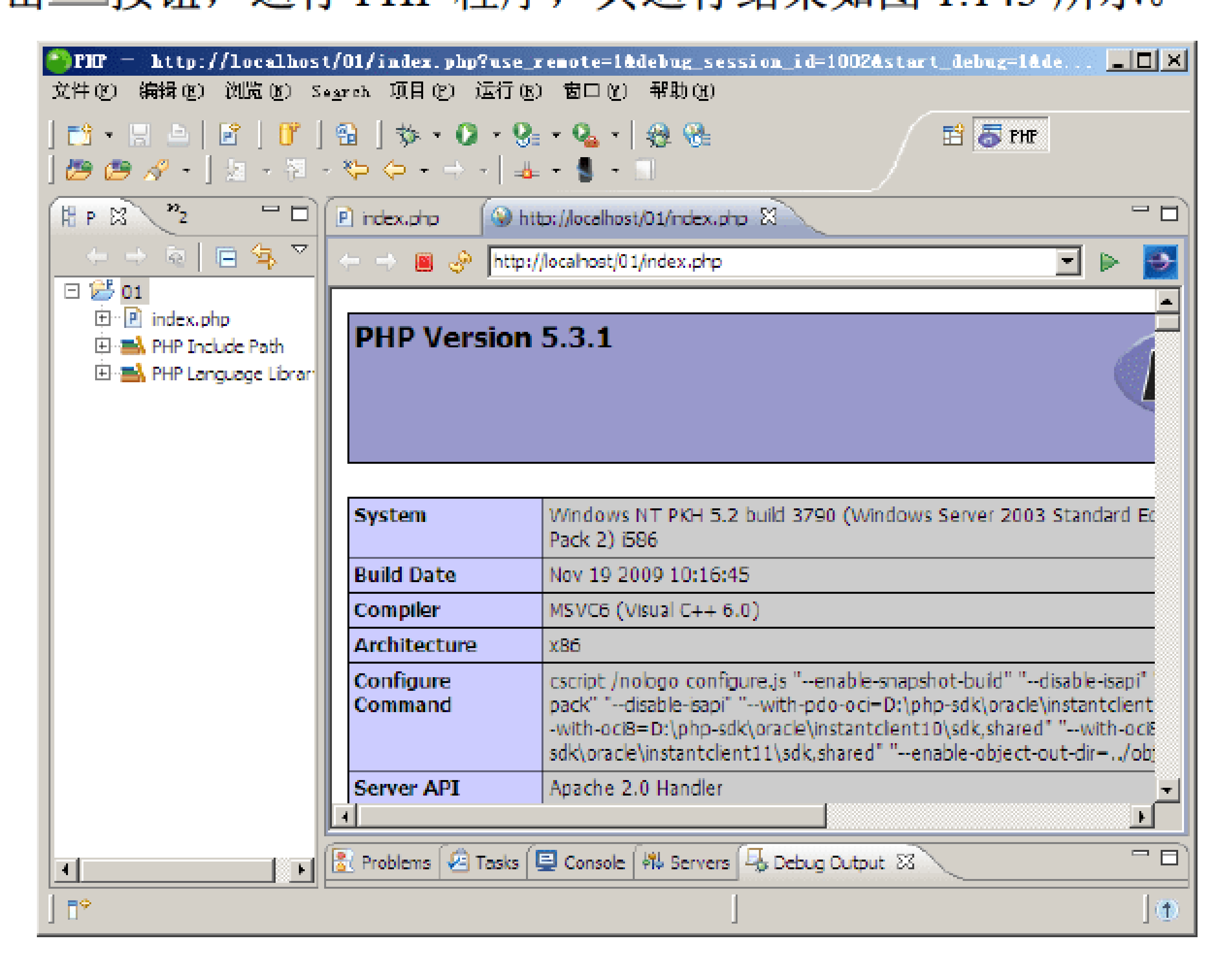
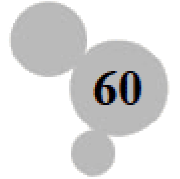


图 1.143　运行程序



关键技术

在 Zend Studio 中部署 Apache 服务器，关键是指定正确的访问路径，如图 1.144 所示。

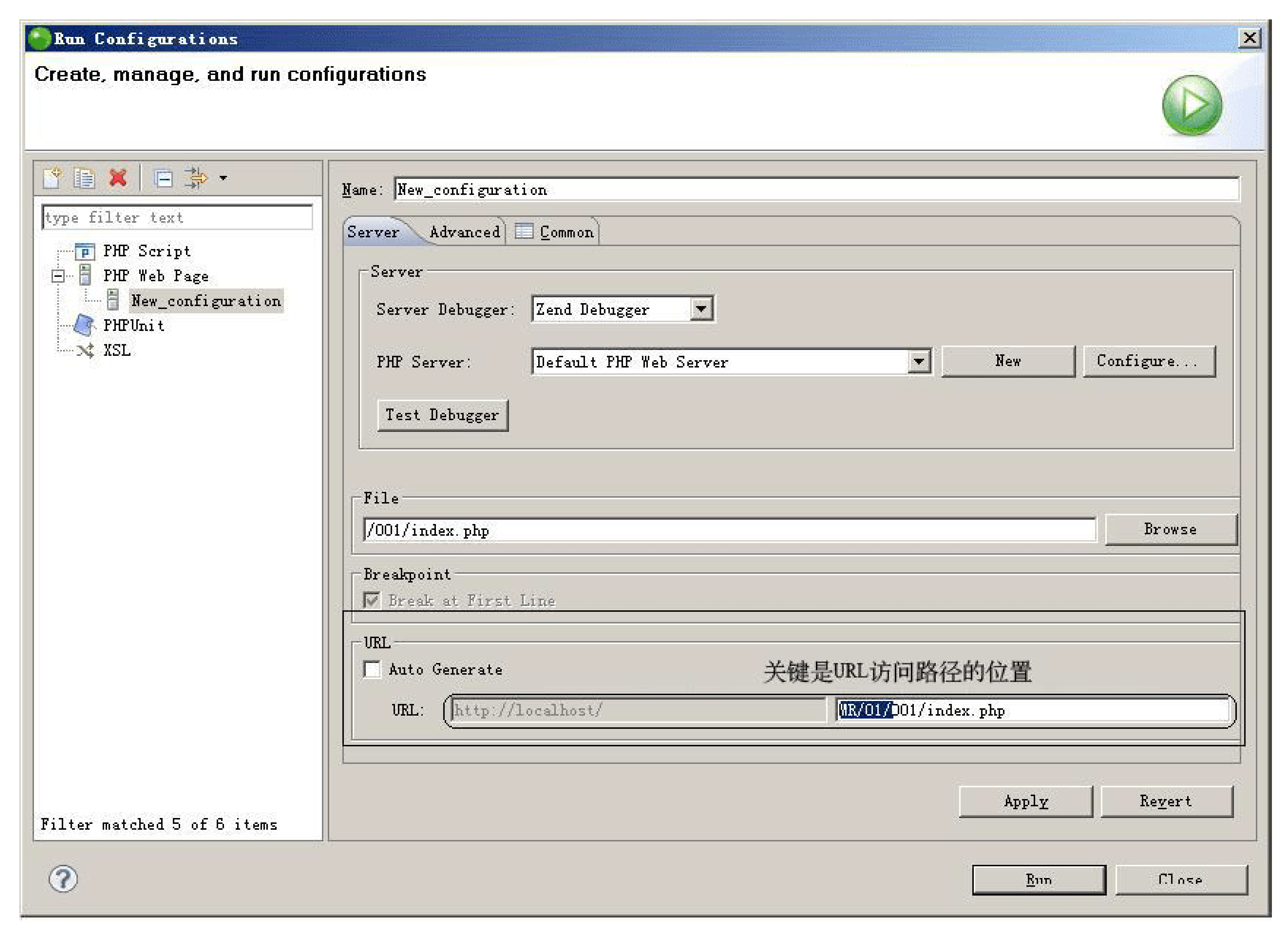


图 1.144　部署 Apache 服务器的访问路径

如果在图 1.144 所示的界面中，没有指定 URL 中访问文件的具体位置 MR/01/001，那么程序就不能正常运行，将出现如图 1.145 所示的效果。



图 1.145　未指定 URL 访问路径

设计过程

在 Zend Studio 中部署 Apache 服务器的操作步骤如下：

（1）创建一个 PHP 项目，这里不再赘述。

（2）单击工具栏中按钮的下三角按钮，将弹出一个下拉菜单，选择 Run Configurations 选项，在弹出的对话框中完成对 Apache 服务器的部署，包括文件的指定，如图 1.146 所示。URL 访问路径的设置，如图 1.147 所示。

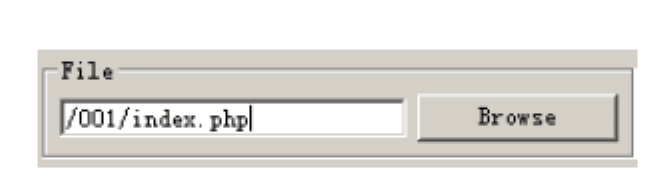


图 1.146　设置访问文件



图 1.147　设置 URL 访问路径

（3）设置完成后单击 Run 按钮，即可浏览 PHP 项目中的内容。

秘笈心法

心法领悟 033：在 Zend Studio 中修改快捷键的设置。

单击 Window 按钮，然后选择 Preferences 命令，将弹出如图 1.148 所示的窗口，可以在其中完成对快捷键的修改操作。

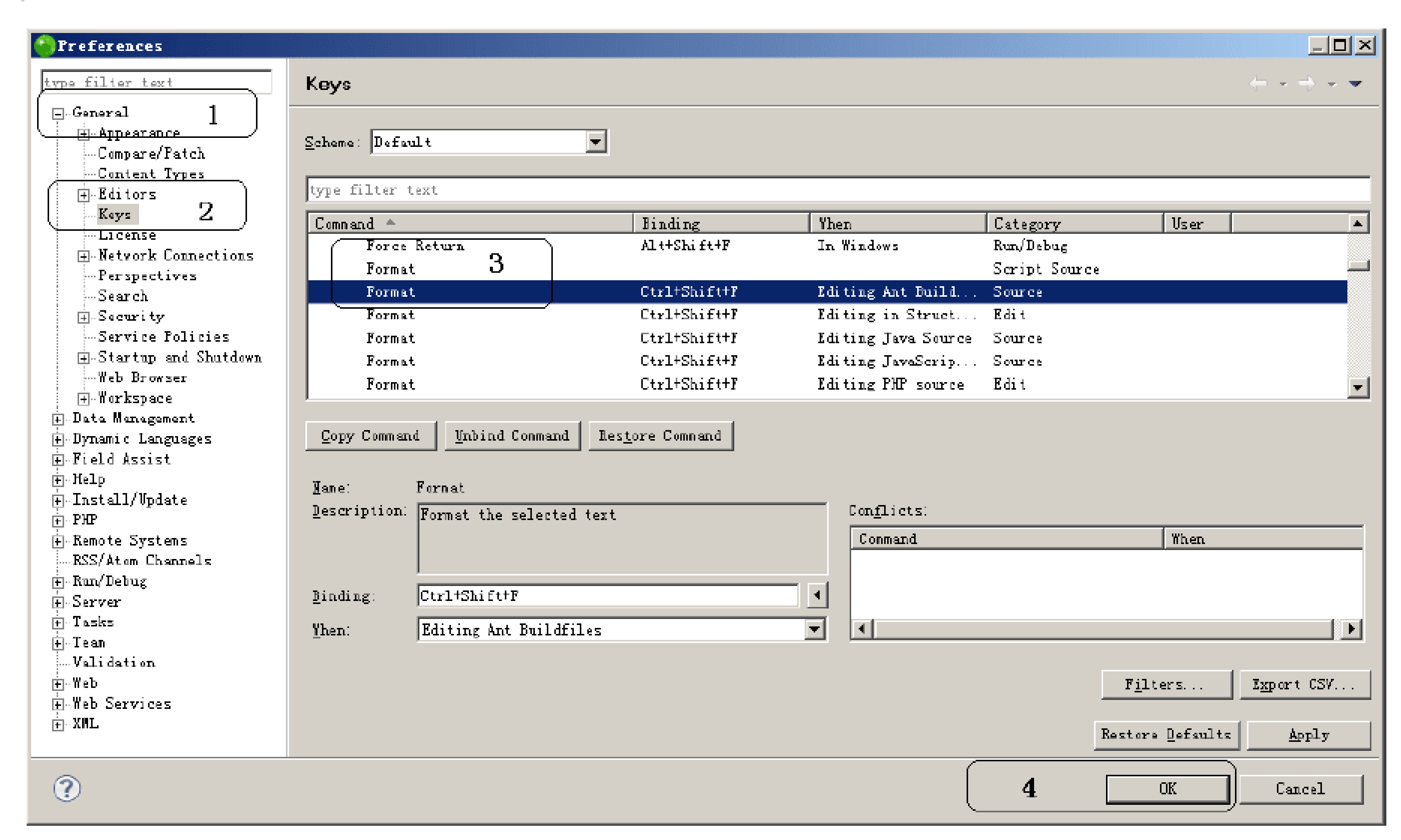


图 1.148　Zend Studio 中快捷键的修改步骤

第 2 章

PHP 基础

- 基本语法
- 运算符
- 包含语句
- 条件语句
- 循环语句
- 跳转语句
- 自定义函数
- 字符串
- 正则表达式
- 数组
- 日期和时间

2.1 基 本 语 法

也许你是一个 PHP 初学者，当你向有经验的 PHP 高级程序员请教如何学好 PHP 这门语言时，想必他会问你有没有其他编程语言的基础，如果有的话，就能够很快入门，学起来也很轻松。

由此可见，掌握一门语言是多么的重要。可以说，扎实的语言基础，是培养和提高程序员编程能力的基本途径，语言基础也是学习 PHP 过程中非常重要的一个环节，它在很大程度上决定了程序员 PHP 学习的进展效果。

另外，程序员的创造力极强，可以完全按照自己的设计目标和思路组织代码，用与其他程序员完全不同的设计方法实现完全相同的功能。但是，编程同时也是一项十分严谨的工作，因为语法是严格限定的，代码必须严格遵守语法要求进行书写，任何不符合规范的代码都不能被计算机执行，哪怕是程序员写错一个字母，或者少写一个结束符都会使程序出错。

所以，程序设计语言的语法必须得到必要的重视。语法是 PHP 语言的规则，打好这个基础，循序渐进地学习，实际运用时才会得心应手。

实例 034 在页面中打印 PHP 的配置信息

光盘位置：光盘\MR\02\034

高级

趣味指数：★★★★

实例说明

本实例应用 PHP 标记，通过 echo 语句和 phpinfo()函数向用户展示 PHP 的配置信息，包括配置文件所在的目录以及一些相关扩展库的版本、作者信息等。运行结果如图 2.1 所示。

PHP Version 5.2.6

System	Windows NT LM 5.2 build 3790
Build Date	May 2 2008 18:01:20
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared" "--with-extra-includes=C:\Program Files (x86)\Microsoft SDK\Include;C:\PROGRA~2\MICROS~2\VC98\ATL\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\MFC\INCLUDE" "--with-extra-libs=C:\Program Files (x86)\Microsoft SDK\Lib;C:\PROGRA~2\MICROS~2\VC98\LIB;C:\PROGRA~2\MICROS~2\VC98\MFC\LIB"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*

图 2.1　PHP 配置信息

关键技术

本实例的关键点是应用 PHP 的输出语句 echo，输出 phpinfo()函数的返回值。phpinfo()函数的代码如下：

```
int phpinfo ( [int what]);
```

该函数将输出 PHP 的配置信息。

设计过程

创建 index.php 文件，应用 PHP 标记，通过 echo 语句输出 phpinfo()函数的返回值，其代码如下：

```
<?php
echo phpinfo();                    //输出 PHP 配置信息
?>
```

秘笈心法

心法领悟 034：phpinfo()函数的具体应用。

在实际的 PHP 项目开发过程中，用户加载的任何一个组件，在 phpinfo()函数的输出页面中都会显示是否已经获得支持，所以此函数的应用对初学者来说是十分重要的。

实例 035	在页面中打印服务器时间 光盘位置：光盘\MR\02\035	高级 趣味指数：★★★★

实例说明

在用 PHP 编写的网站中，经常需要获取当前时间。例如用户在什么时间登录的网站、黑客攻击网站是在什么时间等。本实例应用 PHP 标记，通过 echo 语句和 date()函数来实现获取服务器时间，并进行打印。运行结果如图 2.2 所示。



图 2.2　系统当前时间

关键技术

本实例主要通过 date()函数获取系统的格林威治时间，其语法及说明如下：

```
date(string format,int timestamp);
```

参数 format 指定日期和时间输出的格式，如表 2.1 所示，参数 timestamp 为可选参数，指定时间戳，如果没有指定时间戳，则使用本地时间 time()。

表 2.1　参数 format 的格式化选项

参　数	说　明
a	小写的上午和下午值，返回值为 am 或 pm
A	大写的上午和下午值，返回值为 AM 或 PM
B	Swatch Internet 标准时间，返回值为 000~999
d	月份中的第几天，有前导 0 的 2 位数字，返回值为 01~31
D	星期中的第几天，文本格式，3 个字母，返回值为 Mon~Sun

续表

参　数	说　明
F	月份，完整的文本格式，返回值为 January~December
g	小时，12 小时格式，没有前导 0，返回值为 1~12
G	小时，24 小时格式，没有前导 0，返回值为 0~23
i	有前导 0 的分钟数，返回值为 00~59
I	判断是否为夏令时，如果是夏令时，返回值为 1，否则为 0
j	月份中的第几天，没有前导 0，返回值为 1~31
l	星期数，完整的文本格式，返回值为 Sunday~Saturday
L	判断是否为闰年，如果是闰年，返回值为 1，否则为 0
m	数字表示的月份，有前导 0，返回值为 01~12
M	3 个字母缩写表示的月份，返回值为 Jan~Dec
n	数字表示的月份，没有前导 0，返回值为 1~12
o	与格林威治时间相差的小时数，如 0200
r	RFC 822 格式的日期，如 Thu,21 Dec 2000 16:01:07 +0200
s	秒数，有前导 0，返回值为 00~59
S	每月天数后面的英文后缀，两个字符，如 st、nd、rd 或者 th，可以和 j 一起使用
t	指定月份所应有的天数
T	本机所在的时区
U	从 UNIX 纪元（January 1 1970 00:00:00 GMT）开始至今的秒数
w	星期中的第几天，数字表示，返回值为 0~6
W	ISO-8601 格式年份中的第几周，每周从星期一开始
y	2 位数字表示的年份，如 88、08
Y	4 位数字完整表示的年份，如 1998、2008
z	年份中的第几天，返回值为 0~366
Z	时差偏移量的秒数。UTC 西边的时区偏移量总是负的，UTC 东边的时区偏移量总是正的，返回值为−43200~43200

设计过程

创建 index.php 文件，通过 echo 语句输出 date()函数的返回值，其代码如下：

```
<?php
echo date('Y-m-s H:i:s');                    //输出当前时间
?>
```

秘笈心法

心法领悟 035：对系统时间的设置。

看了上面的实例，读者也许会感到奇怪，为什么系统当前的时间比正常的时间少了 8 个小时？原因是系统输出的当前时间是系统默认的格林威治时间，而不是我们所使用的东八区的北京时间。

想要更改为东八区的时间，有两种方法：

（1）修改 php.ini 文件中的设置，找到[date]下的“;date.timezone = ”选项，将该项修改为“date.timezone =Asia/Hong_Kong”，然后重启 Apache 服务器。



（2）在程序中，通过在使用日期格式之前添加“date_default_timezone_set(timezone);”函数来完成对时区的设置。参数 timezone 为 PHP 可识别的时区名称，如果 PHP 无法识别时区名称，则系统采用 UTC 时区。在 PHP 手册中提供了各时区的名称列表，其中北京时间可以设置时区为 PRC（中华人民共和国）、Asia/Chongqing（重庆）、Asia/Shanghai（上海）、Asia/Urumqi（乌鲁木齐），这几个时间是等效的。

说明： 系统默认的格林威治标准时间（Greenwich Mean Time，GMT）与北京时间相差 8 个小时，在国际无线电通信领域，使用一个统一的时间，称为通用协调时间（Universal Time Coordinated，UTC）。UTC 与格林威治标准时间相同。

设置完成后，时间显示如图 2.3 所示。

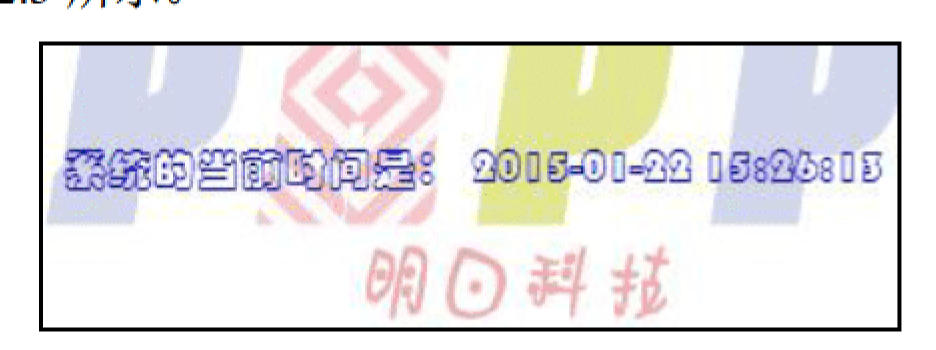


图 2.3　系统设置后的当前时间

实例 036	在页面中打印当前执行的 PHP 文件名 光盘位置：光盘\MR\02\036	高级 趣味指数：★★★★

实例说明

很多时候用户需要编写包含文件路径及文件名称的代码，如果通过目录查找，未免有些麻烦。这时用户可以使用__FILE__预定义函数。本实例通过__FILE__预定义常量获取目标文件的路径及文件名称并在网页上进行打印，运行结果如图 2.4 所示。

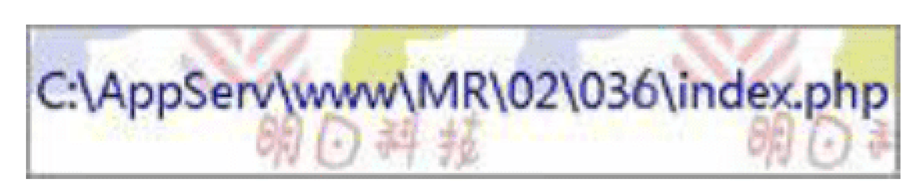


图 2.4　输出当前文件路径和名称

关键技术

系统预定义常量和用户自定义常量在使用上没有差别。大多数预定义常量的执行结果都是服务器的相关信息（版本号、路径、错误参数等），所以程序员很少将此函数用于网站前台的开发，因为如果被别有用心的人知道了这些信息，会严重威胁服务器的安全。使用此函数的语法说明如下：

__FILE__预定义常量：文件的完整路径和文件名。如果用在包含文件中，则返回包含文件名。自 PHP 4.0.2 起，__FILE__ 总是包含一个绝对路径，而在此之前的版本有时会包含一个相对路径。

设计过程

创建 index.php 文件，通过 echo 语句输出__FILE__预定义常量，其代码如下：

```
<?php
echo __FILE__;                              //获取文件的完整路径
?>
```

秘笈心法

心法领悟 036：预定义常量的说明。

PHP 中可以使用预定义常量获取 PHP 中的信息，常用的预定义常量如表 2.2 所示。

表 2.2 PHP 的预定义常量

常 量 名	功 能
__FILE__	默认常量，PHP 程序文件名
__LINE__	默认常量，PHP 程序行数
PHP_VERSION	内建常量，PHP 程序的版本，如 3.0.8_dev
PHP_OS	内建常量，执行 PHP 解析器的操作系统名称，如 Windows
TRUE	该常量是一个真值（TRUE）
FALSE	该常量是一个假值（FALSE）
NULL	一个 NULL 值
E_ERROR	该常量指到最近的错误处
E_WARNING	该常量指到最近的警告处
E_PARSE	该常量指到解析语法有潜在问题处
E_NOTICE	该常量指到不寻常但不一定是错误处

注意：__FILE__和__LINE__中的“__”是两条下划线，而不是一条“_”。表中以 E_开头的预定义常量是 PHP 的错误调试部分。如果想详细了解，请参考 error_reporting()函数。

实例 037 区分单引号和双引号

光盘位置：光盘\MR\02\037

高级

趣味指数：★★★★

实例说明

输出或定义字符串时离不开单引号和双引号的修饰，表面上看它们似乎没有什么不同，实际上是有区别的。本实例应用单引号与双引号操作符来说明如何区分单引号和双引号，运行结果如图 2.5 所示。

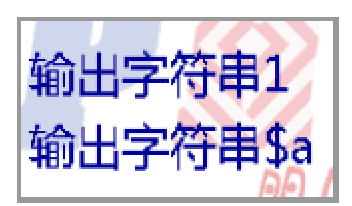


图 2.5 区分单引号和双引号

关键技术

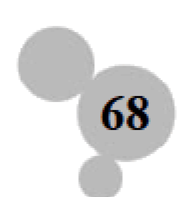
本实例的关键点是输出单引号和双引号修饰的字符串以显示其区别。

使用双引号与单引号最大的区别是，双引号中所包含的变量会自动被替换成实际数值，而在单引号中包含的变量则按普通字符串输出。

设计过程

创建 index.php 文件，通过 echo 语句输出用单、双引号修饰的字符串，其代码如下：

```
<?php
$a = 1;
echo "输出字符串$a"."<br>";
echo '输出字符串$a';
?>
```



秘笈心法

心法领悟 037：转义字符。

通过双引号定义字符串，可以支持很多的转义字符。如表 2.3 所示为双引号支持的转义字符。

表 2.3　双引号支持的转义字符

转 义 字 符	含　　义
\n	换行（LF 或 ASCII 字符 0x0A（10））
\r	回车（CR 或 ASCII 字符 0x0D（13））
\t	水平制表符（HT 或 ASCII 字符 0x09（9））
\\	反斜杠
\$	美元符号
\'	单引号
\"	双引号
\[0-7]{1,3}	此正则表达式序列匹配一个用八进制符号表示的字符，如\467
\x[0-9A-Fa-f]{1,2}	此正则表达式序列匹配一个用十六进制符号表示的字符，如\x9f

\n 和\r 在 Windows 系统中没有什么区别，都可以当作回车符，但在 Linux 系统中则是两种效果。在 Linux 中，\n 表示换到下一行，却不会回到行首；而\r 表示光标会回到行首，但仍然在本行。如果读者使用 Linux 操作系统，可以尝试一下这两个转义字符的应用。

技巧：在定义简单的字符串时，使用单引号是比较合适的处理方式。如果使用双引号，PHP 将花费一些时间来处理字符串的转义和变量的解析。因此，笔者建议在定义字符串时，如果没有特别的要求，应该尽量使用单引号。通过单引号定义的字符串类型变量按照普通字符进行输出。例如，通过单引号定义字符串'$result'，那么将输出'$result'，而不会输出变量$result 的实际值。

说明：（1）在字符串中表示单引号，需要使用反斜线“\”进行转义。例如，定义一个字符串“It's dogged that does it.”，则需要写出“It\'s dogged that does it.”。

（2）在单引号或者字符串结束位置表示反斜线，同样要用反斜线进行转义，即用两个反斜线“\\”表示。例如，表示字符串“F:\PkhPHP\www”，则要写出“F:\\PkhPHP\\www”。

实例 038　动态输出 JavaScript 代码　高级

光盘位置：光盘\MR\02\038　趣味指数：★★★★

实例说明

JavaScript 语言是一门功能强大的客户端脚本语言，也是一门跨平台语言。PHP 支持使用 JavaScript 编码。本实例通过 PHP 5.0 新型字符串动态输出 JavaScript 代码，运行结果如图 2.6 所示。

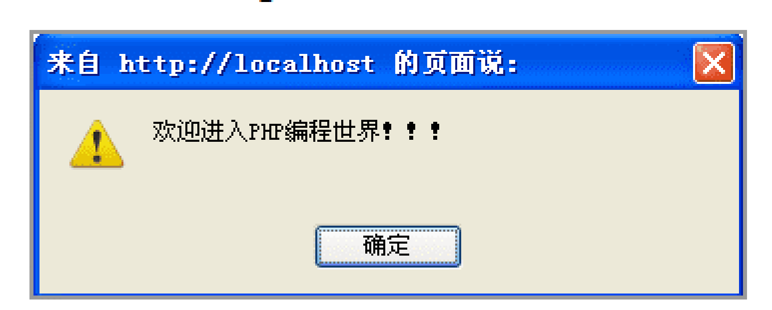


图 2.6　动态输出 JavaScript 代码

关键技术

PHP 5.0 新型字符串的使用是本实例的关键。PHP 5.0 这种新型的字符串是以“<<<”开始，后面紧跟着字符串开始标记，之后为字符串的内容，最后以标记加分号（本实例以 mark 作为结束标记）结束。本函数一般用于对 HTML 代码的格式输出。

设计过程

创建 index.php 文件，通过 PHP 5.0 新型字符串实现实例，其代码如下：

```
<?php
$str=<<<mark                                          //PHP 5.0 新型字符串开始部分
<script language="javascript" type="text/javascript">  //JavaScript 代码
        alert("欢迎进入 PHP 编程世界！！！");           //输出提示
</script>                                             //结束标记
mark;                                                 //新型字符串介绍标记
echo $str;                                            //输出字符串
?>
```

秘笈心法

心法领悟 038：新型字符串。

笔者认为，新型字符串是将以往在 HTML 标签中嵌套 PHP 标签的方式做了一次有效的逆转。

实例 039	当数字遇到了字符串 光盘位置：光盘\MR\02\039	高级 趣味指数：★★★★

实例说明

在 PHP 手册中经常会看到一些返回值为 Boolean 的函数，函数说明的讲解可能是当符合条件时，返回值为 1，否则返回值为 0。按照正常的思维，返回值应该是 TRUE 和 FALSE，因为它们才是真正的 Boolean 类型。其实这些都是类型转换的应用。本实例介绍 PHP 中的数据类型，展示数据是如何进行类型转换的，运行结果如图 2.7 所示。

自动类型转换：
10+我是字符串型数据=10
强制类型转换：
10+我是字符串型数据=10

图 2.7　当数字遇到了字符串

关键技术

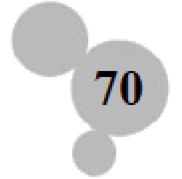
虽然 PHP 是弱类型语言，但有时仍然需要用到类型转换。PHP 中的类型转换和 C 语言一样，非常简单。本实例的关键点是运用类型转换的特点进行输出。

（1）系统根据情况自动转换数据类型称为自动转换。

（2）用户在变量名称前将想要转换的数据类型写到括号内称为强制转换。

设计过程

（1）创建 PHP 脚本，通过 echo 语句输出想要转换的变量，其代码如下：



```
<?php
	$a = 10;				//整型数据
	$b = "我是字符串型数据";		//字符串类型数据
	$e = $a + $b;
	$f = $b + $a;
	echo "自动类型转换：<br>";
	echo '10+我是字符串型数据='.$e."<br>";
	echo "强制类型转换：<br>";
	echo '10+我是字符串型数据='.(string)$e."<br>";
?>
```

（2）将文件存储于\MR\02\039 文件夹中，命名为 index.php。运行结果如图 2.7 所示。

秘笈心法

心法领悟 039：类型转换的相关说明。

（1）转换成整型。在将非整型数据转换成整型时，方法是在变量前使用“(integer)”或者“(int)”，转换规则如下。

☑ 浮点型转换成整型：小数点后的数将被舍弃，如果浮点数超出整数取值范围，那么将无法得到有效的整型结果，结果可能是 0 或者整型的最小负数。

☑ 布尔型转换成整型：TRUE 将转换为 1，FALSE 将转换为 0。

☑ 字符串型转换为整型：将会对字符串左侧的第 1 位进行判断。如果第 1 位是数字，则从第 1 位开始将读取到的数字转换成整型；如果第 1 位不是数字，则结果为 0。

（2）转换成浮点型。方法是在变量前使用“(float)”，转换规则如下：

☑ 整型转换为浮点型：其结果不会发生变化。

☑ 布尔型转换为浮点型：TRUE 将转换为 1，FALSE 将转换为 0。

☑ 字符串型转换为浮点型：如果字符串中包含小数点“.”或科学记数法的 e 或者 E 中的任何一个字符，字符串被当作浮点型处理；否则，被视为整型。

（3）转换成字符串型。方法是在变量前使用“(string)”，转换规则如下。

☑ 整型或者浮点型转换成字符串型：转换结果为其数值。

☑ 布尔型转换为字符串型：TRUE 将转换为字符串“1”，FALSE 将转换为空字符串。

☑ 对象或者数组型转换成字符串型：转换结果为字符串对象或者字符串数组。

☑ 资源型转换成字符串型：转换结果为一个类似“Resource id #”的字符串。在“#”之后是 PHP 在运行时分配给该资源的标识代号。

（4）转换成布尔型。方法是在变量前使用“(boolean)”或者“(bool)”。因为布尔型只包含两个值，TRUE 和 FALSE，所以其转换规则包括两种。

第一种，转换结果为 FALSE 的情况。

☑ 整型或者浮点型数 0。

☑ 空字符串和字符串“0”。

☑ 没有任何元素的空数组。

☑ 没有任何元素的对象。

☑ 特殊类型 NULL。

第二种，转换结果为 TRUE 的情况。排除在第一方面说明的情况，其他转换结果都为 TRUE。

（5）转换成数组。方法是在变量前使用“(array)”，转换规则是：将非数组型转换成与原变量数据类型相同的数组，数组中只有一个元素。

（6）转换成对象。方法是在变量前使用“(object)”，转换规则是：将非对象型转换成一个新的对象，其中名为 scalar 的成员变量将包含原变量的值。

实例 040　PHP 程序员的基础——变量的应用

光盘位置：光盘\MR\02\040

高级

趣味指数：★★★★

实例说明

大楼盖得好不好主要在于地基是否牢固，变量就是 PHP 编程的地基。本实例介绍变量的定义与应用，通过 echo 语句实现打印“Hello World!!!”的小程序，运行结果如图 2.8 所示。



图 2.8　Hello World 程序

关键技术

本实例的关键点是用“$”符号定义变量，通过 echo 语句输出变量。PHP 中的变量使用“$”加变量名来表示，变量名是区分大小写的。使用变量时，只需对变量直接赋值即可。变量赋值，是指给变量一个具体的数据值，对于字符串和数字类型的变量，可以通过“=”来实现，其格式如下：

```
$name = value;
```

设计过程

（1）创建 PHP 脚本文件，定义字符串变量值为“Hello World!!!”，通过 echo 语句输出定义的变量，其代码如下：

```
<?php
	$str = "Hello World!!!";			//定义变量
	echo $str;				//输出变量
?>
```

（2）将该文件存储于\MR\02\040 文件夹下，命名为 index.php。运行结果如图 2.8 所示。

秘笈心法

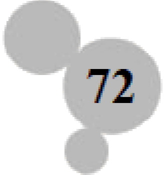
心法领悟 040：变量的作用域。

变量有其自己的作用域，不同的作用域有不同的作用范围，就像清华大学学生的学生证不能在北京大学的图书馆借阅图书一样。

变量按其作用域可以分为全局变量、局部变量和静态变量，变量作用域的说明如表 2.4 所示。

表 2.4　变量作用域的说明

作　用　域	说　　明
全局变量	即被定义在所有函数以外的变量，其作用域是整个 PHP 文件，但是在用户自定义函数内部是不可用的。想在用户自定义函数内部使用全局变量，要使用 global 关键字声明，或者通过使用全局数组$globals 进行访问
局部变量	即在函数的内部定义的变量，这些变量只限于在函数内部使用，不能在函数外部使用
静态变量	能够在函数调用结束后保留变量值，当再次回到其作用域时，又可以继续使用原来的值。而一般变量在函数调用结束后，其存储的数据值将被清除，所占的内存空间被释放。使用静态变量时，先要用关键字 static 来声明变量（需要把关键字 static 放在要定义的变量之前）



注意：在函数的内部定义的变量，其作用域是所在函数。如果在函数外赋值，将被认为是完全不同的另一个变量。在退出声明变量的函数时，该变量及相应的值就会被撤销。

实例 041	打印系统环境变量信息 print_r($_ENV) 光盘位置：光盘\MR\02\041	高级 趣味指数：★★★★

实例说明

PHP 编程语言中预定义了很多有用的变量，这些变量往往起着非常重要的作用。本实例讲解预定义变量 $_ENV 的相关知识，打印输出系统的环境变量，运行结果如图 2.9 所示。

```
Array ( [ALLUSERSPROFILE] => C:\Documents and Settings\All Users
[CLASSPATH] => C:\Program Files\StormII\Codec;C:\Program Files\StormII
[CommonProgramFiles] => C:\Program Files\Common Files [COMPUTERNAME] =>
PC-201006101638 [ComSpec] => C:\WINDOWS\system32\cmd.exe
[FP_NO_HOST_CHECK] => NO [NUMBER_OF_PROCESSORS] => 1 [OS] => Windows_NT
[PATH] => C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
[PATHEXT] => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
[PROCESSOR_ARCHITECTURE] => x86 [PROCESSOR_IDENTIFIER] => x86 Family 15
Model 2 Stepping 9, GenuineIntel [PROCESSOR_LEVEL] => 15
[PROCESSOR_REVISION] => 0209 [ProgramFiles] => C:\Program Files
[SystemDrive] => C: [SystemRoot] => C:\WINDOWS [TEMP] => C:\WINDOWS\TEMP
[TMP] => C:\WINDOWS\TEMP [USERPROFILE] => C:\Documents and
Settings\LocalService [windir] => C:\WINDOWS [AP_PARENT_PID] => 292 )
```

图 2.9　预定义变量的应用

关键技术

PHP 提供的预定义变量有很多是非常实用的，通过这些预定义变量可以获取到用户会话、用户操作系统的环境和本地操作系统的环境等信息，其中$_ENV 是执行环境提交至脚本的变量。

设计过程

（1）创建 PHP 脚本文件，通过 print_r 语句输出系统预定义变量，其代码如下：

```
<?php
    print_r($_ENV);            //打印系统的环境变量
?>
```

（2）将该文件存储于\MR\02\041 文件夹下，命名为 index.php，运行结果如图 2.9 所示。

秘笈心法

心法领悟 041：重要的系统预定义变量。

系统的预定义变量是比较常用的技术，读者必须牢记，尤其是$_FILES、$_POST、$_GET 和$_SESSION。

实例 042	使用可变变量输出“I Like PHP!” 光盘位置：光盘\MR\02\042	高级 趣味指数：★★★★

实例说明

变量包括预定义变量、可变变量等，预定义变量在实例 041 中已经做了简要介绍，本实例通过可变变量实现输出指定字符串，运行结果如图 2.10 所示。

```
I Like PHP!
```

图 2.10　使用可变变量输出

关键技术

可变变量是一种独特的变量，它允许动态改变一个变量名称，其工作原理是该变量的名称由另外一个变量的值来确定。实现过程是在变量前多加一个“$”符号。

设计过程

（1）创建 PHP 脚本文件。首先定义一个字符串变量，然后在第 1 个字符串变量的值前加一个“$”作为第 2 个变量的变量名，最后通过 echo 语句输出可变变量，其代码如下：

```
<?php
    $str_name = "str_name_1";              //定义变量
    $str_name_1 = "I Like PHP!";           //定义可变变量
    echo $$str_name;                       //输出可变变量
?>
```

（2）将该文件存储于\MR\02\042 文件夹下，命名为 index.php。运行结果如图 2.10 所示。

秘笈心法

心法领悟 042：可变变量。

可变变量即是将两个变量的值使用一个变量命名进行操作，但是要注意被引用的变量名前要再加上一个“$”符号。

实例 043	使用转义字符输出特殊字符 光盘位置：光盘\MR\02\043	高级 趣味指数：★★★★

实例说明

在定义长字符串变量时，字符串本身往往包含一些特殊字符。例如，字符串本身包含双引号（"），这时就需要使用转义字符对特殊字符进行转义。本实例通过转义字符“\”转义经特殊字符双引号（"）包含的字符串，运行结果如图 2.11 所示。

"PHP是一门Web开发最火的编程语言"

图 2.11　使用转义字符输出特殊字符

关键技术

本实例的关键点是转义字符“\”的使用，它除了可以做转义字符外，还有一些其他功能，即可以将一些不可打印的字符显示出来，如\a、\b、\e、\f、\n、\r 和\t 等。

设计过程

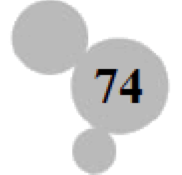
（1）创建 PHP 脚本文件，通过 echo 语句输出经转义字符转义的特殊字符，其代码如下：

```
<?php
    echo "\"PHP 是一门 Web 开发最火的编程语言\"";    //输出转义字符转义的字符串
?>
```

（2）将该文件存储于\MR\02\043 文件夹下，命名为 index.php。运行结果如图 2.11 所示。

秘笈心法

心法领悟 043：转义字符的功能。



转义字符的功能只有一个，就是将特殊字符转义成普通字符。

实例 044　使用常量指定 PI 的值计算圆的面积　高级
光盘位置：光盘\MR\02\044　趣味指数：★★★★

实例说明

常量是 PHP 编程基础的重要组成部分，其作用是定义一个不会改变的值。本实例通过计算圆的面积向用户说明常量是如何定义和使用的，运行结果如图 2.12 所示。

半径为10个单位的圆的面积314.15926

图 2.12　使用常量指定 PI 的值计算圆的面积

关键技术

PHP 中通常使用 define()函数来定义常量；使用 constant()函数动态获取常量值；使用 defined()函数判断一个常量是否已经定义；使用 get_defined_constants()函数获取所有当前已经定义的常量。其中，define()函数的语法如下：

```
bool define ( string name, mixed value [, bool case_insensitive] );
```

参数说明：

string name：必选参数，常量名称，即标识符。

mixed value：必选参数，常量的值。

bool case_insensitive：可选参数，指定大小写是否敏感。设定为 TRUE，表示不敏感。

设计过程

（1）创建 PHP 脚本文件，首先通过 define()函数定义常量，将数值“3.1415926”赋给常量 PI，，然后定义数值型变量，将圆的半径设定为 10 个单位。最后通过 echo 语句输出圆面积，其代码如下：

```
<?php
	define("PI",3.1415926);				//定义常量
	$r = 10;					//定义圆半径
	echo "半径为 10 个单位的圆的面积".PI*($r*$r);	//定义圆面积
?>
```

（2）将该文件存储于\MR\02\044 文件夹下，命名为 index.php。运行结果如图 2.12 所示。

秘笈心法

心法领悟 044：使用 define()函数定义常量的注意事项。

使用 define()函数定义的常量一旦定义就不能改变或者取消。

2.2　运　算　符

运算符是用来对变量、常量或数据进行计算的符号，是对一个值或一组值执行一个指定的操作。PHP 的运算符包括字符串运算符、算术运算符、赋值运算符、递增或递减运算符、位运算符、逻辑运算符、比较运算符和条件运算符。

实例 045 自定义数字的加密/解密算法

光盘位置：光盘\MR\02\045

高级

趣味指数：★★★★

实例说明

运算符是表达式的组成部分，没有运算符的表达式是不存在的。本实例通过使用算术运算符设置数字的加密/解密算法，运行结果如图 2.13 和图 2.14 所示。



图 2.13　口令加密



图 2.14　口令解密

关键技术

本实例的关键点是将数字进行算术运算进而得到加密/解密结果。算术运算（Arithmetic Operators）符号是处理四则运算的符号，在数字处理中应用得最多。

设计过程

（1）创建 PHP 脚本文件，首先定义常量，将数值“3.1415926”赋给常量 PI。然后编写自定义函数，在自定义函数方法体内定义加密或解密算法。最后，当表单提交后，调用对应的加密或解密算法，并输出结果，其代码如下：

```
<?php
    echo "<form action='' method='post'>";                              //输出表单
    echo "数字口令：<input name='text' type='text'>";
    echo "<input type='submit' name='sub' value='确定'>";
    echo "</form>";
    define("PI",3.1415926);                                             //定义常量
    function Encrypt($str){                                             //自定义加密算法
        return $str = $str + PI;                                        //将口令与常量相加
    }
    function Decrypt($str){                                             //自定义解密算法
        return $str = $str - PI;                                        //将加密后口令与常量相减
    }
    if($_POST[sub]){                                                    //通过 POST 方式传参
        echo "加密口令  ".Encrypt($_POST[text])."<br>";         //取得加密后的结果
        $_SESSION[pwd] = Encrypt($_POST[text]);                         //保存在 SESSION 中
?>
    <a href='index.php?pwd=1'>解密口令</a>                                //超链接
<?php
    }
    if(isset($_GET[pwd])){                                              //判断地址栏中是否存在 pwd
        echo "解密口令  ".Decrypt($_SESSION[pwd]);              //口令解密
    }
?>
```

（2）将该文件存储于\MR\02\045 文件夹下，命名为 index.php。运行结果如图 2.13 和图 2.14 所示。

秘笈心法

心法领悟 045：运算符的优先级。

本实例并没有联合应用多种运算符，当联合应用时要注意运算符优先级的问题。所谓的优先级就是先进行哪种运算再进行哪种运算符的运算。为了方便，笔者将运算符的优先级进行归类，如表 2.5 所示。

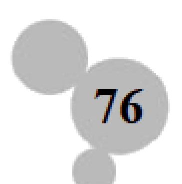


表 2.5　运算符的优先级

优 先 级 别	运　算　符
1	++，--
2	+，－（正、负号运算符），!，~
3	*，/，%
4	+，-
5	<<，>>
6	<，<=，>，>=
7	= =，!=，= = =，!= =，<>
8	&，^，|
9	&&，||
10	?：
11	=，+=，－=，*=，/=，.=，%=
12	and，xor，or

注意：在算术运算符中使用%求余时，如果被除数（$a）是负数，那么取得的结果也是一个负值。

实例 046

比较两个时间戳的大小

光盘位置：光盘\MR\02\046

高级

趣味指数：★★★★

实例说明

在一些程序中，经常需要将两个时间进行比较，但是由于时间是由年、月、日、时、分、秒组成的，比较起来很不方便，这时可以把时间转换成时间戳来进行比较。本实例通过 date()、strtotime()和 ceil()函数实现比较两个时间戳的大小，运行结果如图 2.15 所示。

1421739525 输出日期2015-01-20 15:38:45

1399219200 输出日期2014-05-05 00:00:00

距2014年5月5日已过去261天

图 2.15　比较两个时间戳的大小

关键技术

本实例主要是通过利用 strtotime()函数将时间转换为 UNIX 时间戳，再进行相关取整运算而实现的，其中涉及的相关函数语法如下：

（1）strtotime()函数。该函数预期接受一个包含英文日期格式的字符串，并尝试将其解析为 UNIX 时间戳：

```
int strtotime( string time [, int now]);
```

strtotime()函数有两个参数。如果参数 time 的格式是绝对时间，则 now 参数不起作用；如果参数 time 的格式是相对时间，其对应的时间由参数 now 来提供，若没有提供参数 now，则其对应的时间为当前时间。如果解析失败，则返回 FALSE。在 PHP 5.1.0 之前的版本中，本函数在失败时返回-1。

（2）ceil()函数。该函数返回不小于 value 的下一个整数：

```
float ceil ( float value);
```

设计过程

（1）创建 PHP 脚本文件，首先对 PHP 语言中的时区进行设置，将时区更改为中国上海时区。然后，通过 strtotime()函数获取当前时间的 UNIX 时间戳，并与指定的时间戳进行算术运算。最后，将时间戳运算结果转换为天数，并通过 ceil()函数进行取整运算，然后输出结果，其代码如下：

```
<?php
    date_default_timezone_set("Asia/ShangHai");            //将格林威治时间设置为本地时间
    $a = strtotime("now");                                 //取得当前时间戳
    $b = strtotime("05 May 2014");                         //取得 2014 年 5 月 5 日的时间戳
    echo $a."\n";
    echo "输出日期".date("Y-m-d H:i:s",$a)."<br><br>";     //取得以$a 为时间点的时间
    echo $b."\n";
    echo "输出日期".date("Y-m-d H:i:s",$b)."<br><br>";     //取得以$b 为时间点的时间
    $c = ceil(($a - $b)/(3600*24));                        //计算相差天数
    echo "距 2014 年 5 月 5 日已过去".$c."天";
?>
```

（2）将该文件存储于\MR\02\046 文件夹下，命名为 index.php。运行结果如图 2.15 所示。

秘笈心法

心法领悟 046：时间戳的相关说明。

有效的时间戳的范围通常从 Fri, 13 Dec 1901 20:45:54 GMT 到 Tue, 19 Jan 2038 03:14:07 GMT（对应于 32 位有符号整数的最小值和最大值）。不是所有的平台都支持负的时间戳，因此日期范围就被限制为不能早于 UNIX 纪元。这意味着在 1970 年 1 月 1 日之前的日期将不能用在 Windows、一些 Linux 版本以及几个其他的操作系统中。

注意： 在 UNIX 系统中，日期与时间表示为自 1970 年 1 月 1 日零点起到当前时刻的秒数，这种时间称为 UNIX 时间戳，用 32 位二进制表示。其中，1970 年 1 月 1 日零点称为 UNIX 世纪元。UNIX 时间戳提供了一种统一、简洁的时间表示方式，在不同的操作系统中均受到支持，同一时间在 UNIX 和 Windows 中均以相同的 UNIX 时间戳表示，所以不需要在不同的系统中进行转换。同时，UNIX 时间戳是一个时间差，与时区没有关系，无论当前 PHP 中使用的是何种时区，其 UNIX 时间戳是唯一的。

实例 047 使用条件运算符判断数字的奇偶性

光盘位置：光盘\MR\02\047

高级

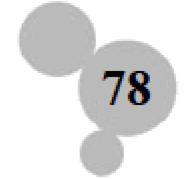
趣味指数：★★★★

实例说明

条件运算符用于执行或真或假运算。本实例演示如何用三元运算符判断数字的奇偶性，运行结果如图 2.16 所示。

0是偶数 1是奇数 2是偶数 3是奇数
5是奇数 6是偶数 7是奇数 8是偶数 9是奇数

图 2.16　用三元运算符判断数字的奇偶性



关键技术

本实例使用三元运算符判断数字的奇偶性，并且在判断数字奇偶性时使用了求余运算符。当期望数字与数字 2 做除法运算时，如果存在余数即不能被 2 整除，为奇数，否则为偶数。三元运算符的语法如下：

```
(expr) ? {statement1;} : {statement2;}
```

三元运算符与 if…else…语句实现的功能完全相同。

设计过程

（1）创建 PHP 脚本文件，首先，通过 for 循环语句循环输出 0~9 的 10 个数字。然后，将得到的 10 个数字分别与 2 做求余运算并将运算结果作为三元运算的条件进行判断。最后，将判断的结果利用 echo 语句输出，其代码如下：

```
<?php
      //做小于 10 的循环
      for($a = 0;$a < 10;$a++){
      //用三元运算输出，如果没有余数则是偶数，否则为奇数
      echo   $a % 2 == 0 ?   $a."是偶数"."\n" : $a."是奇数"."\n";
    }
?>
```

（2）将该文件存储于\MR\02\047 文件夹中，命名为 index.php。运行结果如图 2.16 所示。

秘笈心法

心法领悟 047：用 if…else…语句实现实例。

三元运算等同于 if…else…语句，下面通过 if…else…语句实现本实例，其语法如下：

```
<?php
    //做小于 10 的循环
    for($a = 0;$a < 10;$a++){
      //用 if…else…语句实现
      $b = 0;
      if($a% 2 ==0){
              //设定条件，让$a 与 2 做求余运算
            echo "$a".'是偶数';
      }else{
            echo "$a".'是奇数';
            }
      }
?>
```

实例 048　判断用户是否具有后台管理权限

光盘位置：光盘\MR\02\048

高级　趣味指数：★★★★

实例说明

逻辑运算符往往作为 if 等语句的条件出现。逻辑运算符有很多种而且功能各不相同。本实例介绍逻辑运算符的语法等相关知识，并通过逻辑运算符来演示用户是否具有后台管理权限，运行结果如图 2.17 和图 2.18 所示。



图 2.17　具有后台管理权限

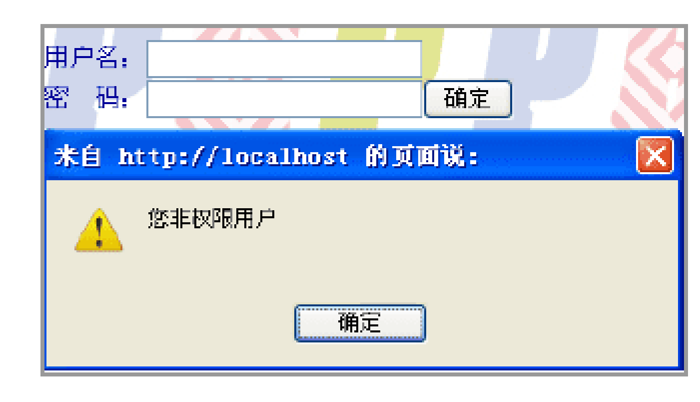


图 2.18　不具有后台管理权限

关键技术

本实例主要是通过逻辑运算符&&（逻辑与），将从表单中利用 POST 方法取得的由文本框输入的用户名和密码联系起来。用户名和密码必须同时为真，if 语句的结果才会为真，反之为假。

设计过程

（1）创建 PHP 脚本文件，首先，编写 form 表单，将表单中的 method 方法定义为 POST。然后，当单击按钮时，将 POST 方法接收的用户名和密码进行逻辑与运算。当 if 语句的结果为 TRUE 时，输出“您具有管理员权限”的提示，否则输出“您非权限用户”的提示，其代码如下：

```
<?php
	echo "<form action="method='post'>";							//输出表单
	echo "用户名：<input type='text' name='text'><br>";
	echo "密  码：<input type='password' name='pwd'>";
	echo "<input type='submit' name='sub' value='确定'>";
	echo "</form>";
	if($_POST[sub]){									//通过 POST 方式提交
		if($_POST[text] == "mr" && $_POST[pwd] == "mrsoft"){		//验证用户名和密码
			echo "<script>alert('您具有管理员权限');</script>";		//符合条件输出提示
		}else{
			echo "<script>alert('您非权限用户');</script>";			//不符合条件输出提示
		}
	}
?>
```

（2）将该文件存储于\MR\02\048 文件夹中，命名为 index.php。运行结果如图 2.17 和图 2.18 所示。

秘笈心法

心法领悟 048：使用优先级的小技巧。

如果在涉及运算符的优先级时不想总查找优先级表而本身又记忆不清的话，建议使用小括号把想要先运算的表达式括起来，这样问题就迎刃而解了。

实例 049	打印随机组合生日祝福语 光盘位置：光盘\MR\02\049	高级 趣味指数：★★★★

实例说明

在应用字符串时往往需要将两个字符串进行连接，这时可以使用字符串运算符。本实例使用字符串运算符打印随机组合生日祝福语，运行结果如图 2.19 所示。

生日快乐祝你万事如意

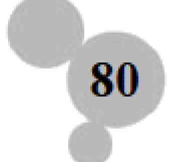
图 2.19　打印随机组合生日祝福语

关键技术

字符串可以用“.”（点）字符串连接符连接，该连接符可以把两个或两个以上字符串连接成一个新的字符串。字符串的连接有两种形式，第一种是连接运算符“.”；第二种是连接赋值运算符“.=”。在实际的程序开发中，这两种连接方式都很常用。

设计过程

创建 index.php 文件，首先，定义两个数组，将字符串信息保存在数组中。然后，利用 rand()函数随机取得



数组中的两条信息并保存在变量中。最后，通过 echo 语句输出用连接运算符连接的变量，其代码如下：

```
<?php
	$arr = array("生日快乐","今天是你的出生日","同学们为你许愿");	//定义数组
	$array = array("祝你万事如意","祝你生日快乐","祝你福如东海长流水寿比南山不老松");	//定义数组
	$rand = rand(0,2);	//定义随机数
	echo $arr[$rand].$array[$rand];	//输出字符串
?>
```

秘笈心法

心法领悟 049：“+”号的相关说明。

使用过 C 或 Java 语言的读者应该注意，这里的“+”号，只做算术运算符使用，而不能做字符串运算符。

实例 050　打印 2000~2020 年间的所有闰年

光盘位置：光盘\MR\02\050

高级

趣味指数：★★★★

实例说明

在 PHP 语言中，为了提升编程效率设置了一些自增自减运算符，这些运算符在循环语句中得到广泛应用。本实例简要介绍自增自减运算符，并通过自增自减运算符修饰的变量来演示其应用，运行结果如图 2.20 所示。



图 2.20　打印 2000~2020 年间的所有闰年

关键技术

递增或递减运算符有两种使用方法，一种是先将变量增加或减少 1 后再将值赋给原变量，称为前置递增或递减运算符（也称前置自增自减运算符）；另一种是将运算符放在变量后面，即先返回变量的当前值，然后变量的当前值增加或减少 1，称为后置递增或递减运算符（也称后置自增自减运算符）。本实例将递增或递减运算符应用到 for 循环的变量中，利用变量的变化完成期望次数的循环。

设计过程

创建 index.php 文件，首先，利用 for 循环定义循环变量，将变量的初始值设置为 2000，最大值小于 2020，并且要求变量做自增运算。然后，设置循环变量与 4 做求余运算。最后，把符合条件的数值通过 echo 语句输出，代码如下：

```
<?php
if($_POST[sub]){	//通过 POST 接收参数
	for($a = 2000;$a <= 2020;$a++){	//通过循环定义循环变量
		if($a % 4 == 0){	//将循环变量与 4 做求余运算
			echo $a."  ";	//输出循环变量
		}
	}
}
?>
```

秘笈心法

心法领悟 050：自增自减运算符。

自增自减运算符在 PHP 循环结构中，特别是 for 循环结构中有着非常重要的地位。至于自增自减运算符前

置和后置的区别，请详见实例 051。

实例 051　前置运算符和后置运算符的区别

光盘位置：光盘\MR\02\051

高级　趣味指数：★★★★

实例说明

前置运算符与后置运算符的应用方向与实例 050 相同，大多用在循环语句中。本实例应用自增自减运算符来修饰变量，通过显示结果来说明前置运算符与后置运算符的区别，运行结果如图 2.21 所示。

----------后置加运算---------
0 1 2 3 4 5 6 7 8 9 10
----------前置加运算---------
1 2 3 4 5 6 7 8 9 10 11
----------前置减运算---------
9 8 7 6 5 4 3 2 1 0 -1
----------后置减运算---------
10 9 8 7 6 5 4 3 2 1 0

图 2.21　前置运算符与后置运算符的区别

关键技术

本实例的关键点是自增自减运算符的灵活运用，相关说明详见实例 050。在使用前置减运算符时，要注意给定的初始值，否则会出现本例的问题，即前置减运算输出负数。

设计过程

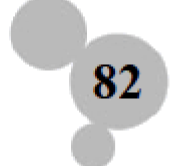
（1）创建 PHP 脚本文件，首先通过 for 循环语句定义循环变量，然后通过 echo 语句输出循环变量，其代码如下：

```
<?php
    $a = 0;                                         //自定义变量并给定初始值
    echo "----------后置加运算---------<br>";
    for($b = 0;$b <= 10;$b++){                      //使用 for 循环语句
        echo   $a++." ";                            //使用后置运算符
    }
    echo "<br>----------前置加运算---------<br>";
    $d = 0;
    for($b = 0;$b <= 10;$b++){
        echo ++$d." ";                              //使用前置运算符
    }
    echo "<br>----------前置减运算---------<br>";
    $f =10;
    for($b = 10;$b >= 0;$b--){
        echo --$f." ";                              //使用前置运算符
    }
    echo "<br>----------后置减运算---------<br>";
    $g =10;
    for($b = 10;$b >= 0;$b--){
        echo $g--." ";                              //使用后置运算符
    }
?>
```

（2）将该文件存储于\MR\02\051 文件夹下，命名为 index.php。运行结果如图 2.21 所示。

秘笈心法

心法领悟 051：前置运算符与后置运算符。



前置运算符是在程序第一次运行时就进行自增或自减运算，而后置运算符是在程序运行完一次之后再进行自增或自减运算。

实例 052　使用位运算符对数字进行加密和解密

光盘位置：光盘\MR\02\052

高级

趣味指数：★★★★

实例说明

位运算符也是 PHP 运算符中不可或缺的一种。本实例通过位运算符实现对数字进行加密和解密，运行结果如图 2.22 和图 2.23 所示。



图 2.22　使用位运算符对数字进行加密

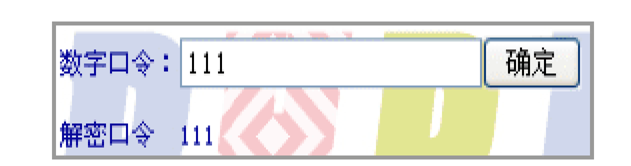


图 2.23　使用位运算符对数字进行解密

关键技术

位运算符是指对二进制位从低到高位对齐后进行运算，PHP 中的位运算符如表 2.6 所示。本实例中，利用自定义函数将文本框信息与定义的常量做指定的位运算，并返回计算结果，从而将文本框信息进行加密或解密处理。

表 2.6　位运算符

符　　号	作　　用	举　　例
&	按位与	$m&$n
|	按位或	$m\|$n
^	按位异或	$m^$n
~	按位取反	$m~$n
<<	向左移位	$m<<$n
>>	向右移位	$m>>$n

设计过程

（1）创建 PHP 脚本文件，首先，编写 form 表单并定义常量，将数值“3.1415926”赋给常量 PI。然后，定义自定义函数，在函数体内部，将文本框信息与常量 PI 做移位运算并返回结果。最后，通过提交数据和单击按钮实现输出加密、解密算法的结果，其代码如下：

```
<?php                                                           //定义表单
    echo "<form action='' method='post'>";
    echo "数字口令：<input name='text' type='text'>";
    echo "<input type='submit' name='sub' value='确定'>";
    echo "</form>";
    define("PI",3.1415926);                                     //定义常量
    function Encrypt($str){                                     //自定义加密算法
        return $str = $str << PI;
    }
    function Decrypt($str){                                     //自定义解密算法
        return $str = $str >> PI;
    }
    if($_POST[sub]){                                            //通过 POST 方式传参
        echo "加密口令  ".Encrypt($_POST[text])."<br>";  //输出口令
```

```
            $_SESSION[pwd] = Encrypt($_POST[text]);                    //将口令保存在 SESSION 中
?>
            <a href='index.php?pwd=1'>解密口令</a>                     //超链接
<?php
      }
      if(isset($_GET[pwd])){                                           //判断地址栏是否存在此参数
            echo "解密口令  ".Decrypt($_SESSION[pwd]);       //输出解密结果
      }
?>
```

（2）将该文件存储于\MR\02\052 文件夹下，命名为 index.php。运行结果如图 2.22 和图 2.23 所示。

秘笈心法

心法领悟 052：位运算符的相关说明。

位运算符允许对整型数据中指定的位进行置位。如果左右参数都是字符串，则位运算符将操作字符的 ASCII 值。

2.3 包含语句

引用外部文件可以减少代码的重用性，是 PHP 编程的重要技巧。PHP 提供了 4 个非常简单有用的语句，它们允许重新使用任何类型的代码。使用任意一个语句均可以将一个文件载入 PHP 脚本中，为网站创建统一风格的页面，从而更加轻松地编写和维护代码。

实例 053	提高代码重用率 光盘位置：光盘\MR\02\053	高级 趣味指数：★★★★

实例说明

当需要在文件外部引入新文件时就要借助 include 或 require 等包含语句。本实例通过 require 包含语句实现代码重用，运行结果如图 2.24 所示。

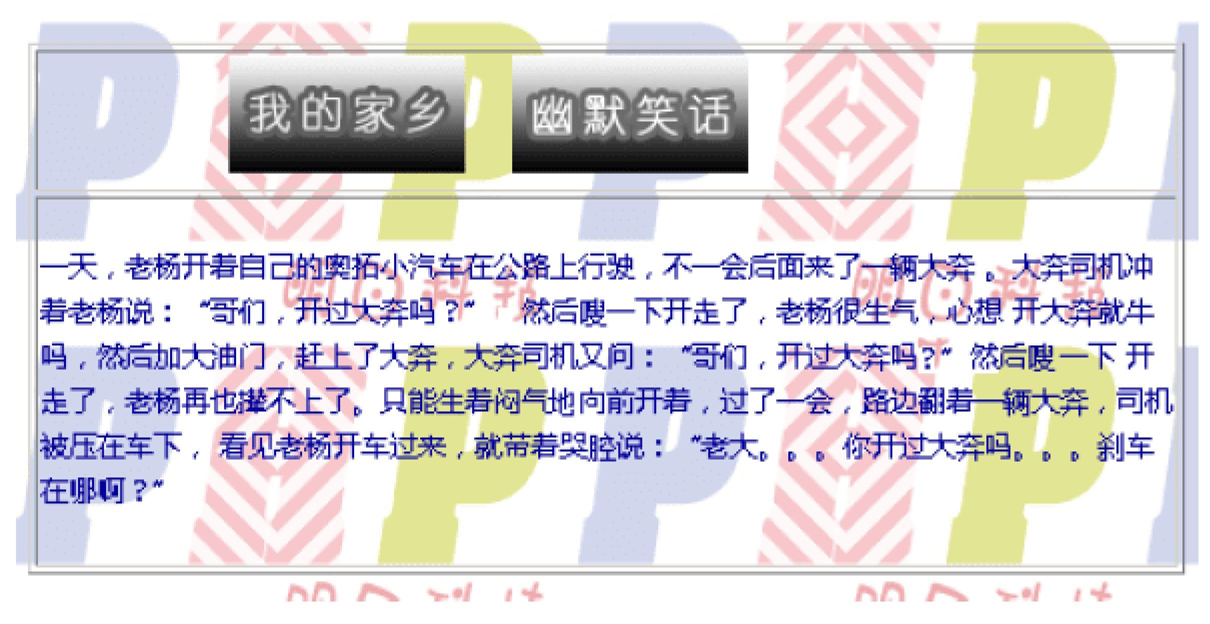
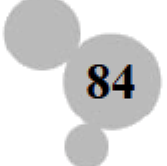


图 2.24　提高代码重用率

关键技术

本实例通过 require 语句包含并运行指定文件。PHP 文件被执行之前，PHP 解析器会用被引用文件的全部内容替换 require 语句，然后与 require 语句之外的其他语句组成新的 PHP 文件。最后再按新的 PHP 文件执行程序代码，其语法如下：

```
void require(string filename);
```



参数说明：

filename：是指定的完整路径文件名。

设计过程

（1）创建 inc.php 文件，建立两个字符串表达式，代码如下：

```
<?php
$str1 ="大连是一个美丽的地方，它三面环海，                          //声明两个字符串变量
背依大山，就像一个乘风破浪的水手，
就算再大的风浪，也依然昂首挺胸不屈不挠";
$str2 = "一天，老杨开着自己的奥拓小汽车在公路上行驶，不一会后面来了一辆大奔。
大奔司机冲着老杨说：“哥们，开过大奔吗？”然后嗖一下开走了，老杨很生气，心想
开大奔就牛吗，然后加大油门，赶上了大奔，大奔司机又问：“哥们，开过大奔吗?”然后嗖一下
开走了，老杨再也撵不上了。只能生着闷气地向前开着，过了一会，路边翻着一辆大奔，司机被压在车下，
看见老杨开车过来，就带着哭腔说：“老大。。。你开过大奔吗。。。刹车在哪啊？”";
?>
```

（2）创建 index.php 文件，通过 require 语句包含文件 inc.php，代码如下：

```
<?php
        require("inc.php");
        switch($_GET['link']){                              //接收地址栏的值
                case "我的家乡";                            //如果值等于“我的家乡”
                echo $str1;                                 //输出字符串 1
                break;                                      //停止执行语句
                case "幽默笑话";                            //如果值等于“幽默笑话”
                echo $str2;                                 //输出字符串 2
                break;                                      //停止执行语句
        }
?>
```

秘笈心法

心法领悟 053：使用 require 语句的注意事项。

因为使用 require 语句相当于将另一个源文件的内容完全复制到本文件中，所以一般将该语句放在源文件的起始位置，用于引用需要使用的公共函数和公共类文件等。

实例 054　包含数据库连接文件

光盘位置：光盘\MR\02\054

高级

趣味指数：★★★★

实例说明

一个程序可能要与数据库多次交互，所以数据库连接信息要单独保存在某一文件中，这也是代码重用的一种体现。本实例通过 include 语句包含数据库文件，运行结果如图 2.25 所示。



图 2.25　包含数据库连接文件

关键技术

include 语句与 require 语句在作用上是完全相同的，但是使用 include 语句引用外部文件时，只有代码执行到该语句时才将外部文件引用进来并读取文件的内容，当所引用的外部文件发生错误时，系统只给出一个警告，

整个 PHP 文件则继续向下执行。include 语句的语法如下：

```
void include(string filename);
```

参数说明：

filename：是指定的完整路径文件名。

设计过程

（1）创建 index.php 文件，通过 include 语句包含数据库文件，并执行查询操作输出结果，其代码如下：

```
<?php
include ("inc.php");                                    //包含数据库连接文件
$sql = "select * from tb_include";                      //SQL 语句
$rs = mysql_query($sql);                                //执行查询操作
while($rst = mysql_fetch_array($rs)){                   //将查询结果循环输出
        echo '编号：'.$rst[0].'  姓名：'.$rst[1]."<br>";
}
?>
```

（2）创建数据库连接文件 inc.php，其代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111");        //连接 MySQL
mysql_select_db("db_database02",$conn);                 //连接数据库
mysql_query("SET NAMES GBK");                           //设置编码格式
?>
```

秘笈心法

心法领悟 054：使用 include 语句的注意事项。

include 语句包含一个不存在的文件时会产生警告。

实例 055	包含网站头文件 光盘位置：光盘\MR\02\055	高级 趣味指数：★★★★

实例说明

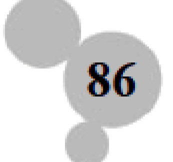
在设计网站的过程中往往都是先用<table>或<div>标签将整个网站进行布局，然后通过 include_once 等语句将各部分内容包含进来。本实例介绍使用 include_once 语句包含网站头文件，运行结果如图 2.26 所示。



图 2.26　包含网站头文件

关键技术

PHP 程序开发中，在使用 include_once 语句时，应该明确其与 include 语句的区别。include_once 语句会在导入文件前检测该文件是否在该页面的其他部分被引用过，如果有，则不会重复引用该文件，即程序只引用一次。例如，要导入的文件中存在一些自定义函数，如果在同一个程序中重复导入这个文件，在第二次导入时便会发生错误，因为 PHP 不允许相同名称的函数被重复声明。



设计过程

（1）创建 index.php 文件，通过 include_once 语句包含文件 inc.php，其代码如下：

```
<table><tr><td><?php include_once("inc.php");?></td></tr></table>                //在 table 标签中嵌入 PHP 包含语句
```

（2）创建 inc.php 文件，将图片作为<body>标签的背景图，其代码如下：

```
<title>包含头文件</title>
<body background="../image/1.jpg" style="font-size:14px; color:#000099; font-family:'微软雅黑'">    //背景
<img src="image/1.jpg">                                                             //引入图片
</body>
```

（3）运行本实例，运行结果如图 2.26 所示。

秘笈心法

心法领悟 055：使用 include_once 语句的注意事项。

include_once 语句应用于在脚本执行期间同一个文件有可能被包含超过一次的情况下，确保指定文件只被包含一次，从而避免函数重复定义，变量重新赋值。

实例 056　包含网站尾文件

光盘位置：光盘\MR\02\056

高级

趣味指数：★★★★

实例说明

require 语句和 include 语句在作用上没有本质区别，用户可根据个人习惯选择使用。本实例通过使用 require 语句包含网站尾文件，运行结果如图 2.27 所示。



图 2.27　包含网站尾文件

关键技术

本实例的关键点是 rerquire 语句的运用。require 语句的详细说明请参见实例 053。

设计过程

（1）创建 index.php 文件，编写 PHP 标记，通过 require 语句包含文件 inc.php，其代码如下：

```
<?php
    require('inc.php');                                                             //包含尾文件
?>
```

（2）创建 inc.php 文件，定义网站尾文件，利用<p>标签将明日科技公司的相关信息包含进来，其代码如下：

```
<table width="800" height="129" border="1" bordercolor="#CCCCCC" bordercolorlight="#CCCCCC" cellspacing="0">
  <tr>
    <td width="800"><div align="center">                                           //定义表格
      <p>长春市明日科技有限公司</p>
      <p>电话：0431-84978981</p>
      <p>传真：0431-84978981</p>
      <p>email:www.mrsoft.com</p>
    </div></td>
```

```
    </tr>
</table>
```

（3）运行本实例，运行结果如图 2.27 所示。

秘笈心法

心法领悟 056：使用 require 语句的注意事项。

require 语句包含不存在的文件时会出现一个致命错误。

实例 057　包含网站主文件

光盘位置：光盘\MR\02\057

高级

趣味指数：★★★★

实例说明

本实例通过使用 require_once 语句包含网站主文件，运行结果如图 2.28 所示。



图 2.28　包含网站主文件

关键技术

require_once 语句是 require 语句的延伸，其功能与 require 语句基本类似，不同的是在应用 require_once 语句时会先检查要引用的文件是不是已经在该程序中的其他地方被引用过，如果有，则不再重复调用该文件.例如，同时应用 require_once 语句在同一个页面中引用了两个相同文件，那么在输出时只有第一个文件被执行，第二次引用的文件不会被执行。

设计过程

（1）创建 index.php 文件，通过 switch 语句接收地址栏传递的参数，根据参数传递的数据引入通过 require_once 语句包含的文件，其代码如下：

```
<?php
    switch($_GET['link']){                          //通过 GET 方式接收地址栏参数
        case "首页":
        require_once('inc.php');                    //包含文件 inc.php
        break;
        case "注册":
```

```
            require_once('inc_1.php');              //包含文件 inc_1.php
            break;
            case "浏览":
            require_once('inc_2.php');              //包含文件 inc_2.php
            break;
            default:
            include('inc.php');
    }
?>
```

（2）运行本实例，运行结果如图 2.28 所示。

秘笈心法

心法领悟 057：使用 require_once 语句的注意事项。

require_once 语句与 require 语句一样，当包含不存在的文件时，会出现一个致命错误。

2.4　条 件 语 句

条件控制语句就是以一定的条件作为依据，根据判断的结果确定执行哪一部分代码，而不执行与该部分代码并列的其他代码。

在 PHP 中，条件控制语句可分为以下两种类型：

☑　if 条件控制语句。

☑　switch...case 分支控制语句。

实例 058	员工生日提醒 光盘位置：光盘\MR\02\058	高级 趣味指数：★★★★

实例说明

在 PHP 应用中，类似生日提醒等功能的程序随处可见，主要是因为这样的程序有一定的定时效果。本实例通过 if 语句和 foreach 循环语句实现一个员工生日提醒的小程序，如图 2.29 所示。



图 2.29　员工生日提醒

关键技术

本实例的关键点是利用 if 条件语句，将文本框中的日期与数组中员工生日日期进行比较。

（1）foreach 语句

foreach 语句仅能用于数组，当试图将其用于其他数据类型或者一个未初始化的变量时会产生错误。

```
foreach (array_expression as $value) statement;
```

参数 array 表示要遍历的数组，expression 表示键值，$value 表示键值的对应值，statement 表示语句块。

（2）if 语句

if 语句对某段程序的执行附加一个条件，如果条件成立，就执行这段程序；否则跳过这段程序。

```
if(expr) statement;
```

参数 expr 表示判定条件，statement 表示语句块。

（3）array()函数

array()函数返回根据参数建立的数组。参数可以用 => 运算符给出索引。

```
array array ( [mixed ...]);
```

设计过程

（1）创建 index.php 文件。首先，根据数组的特点将键值与对应值存储在数组中，然后，利用 foreach 语句遍历数组，将取得的 value 值与当前日期进行比较。最后，将符合条件的值进行输出，其代码如下：

```
<?php
    $a = array("小杨"=>"05 月 30 日","小王"=>"05 月 28 日","小张"=>"05 月 29 日");     //定义数组
        foreach($a as $key => $value){                                              //用 foreach 语句循环数组
            if(date("m 月 d 日") == $value){                                        //用 if 条件语句进行比较
                echo "  今天是".$key."的生日";                            //输出结果
            }
        }
?>
```

（2）运行本实例，运行结果如图 2.29 所示。

秘笈心法

心法领悟 058：使用 foreach 语句的相关说明。

foreach 语句所操作的数据是指定数组的一个副本，而不是该数组本身。因此数组指针不会被 foreach 结构改变，对返回的数组单元的修改也不会影响源数组，不过源数组的内部指针的确在处理数组的过程中向前移动了。假定 foreach 循环运行到结束，源数组的内部指针将指向数组的结尾。foreach 语句不支持用“@”来禁止错误信息。

实例 059	考试成绩评定标准 光盘位置：光盘\MR\02\059	高级 趣味指数：★★★★

实例说明

本实例通过 if 条件语句对文本框中输入的考试成绩进行评定和选择，运行结果如图 2.30 所示。



图 2.30　考试成绩评定标准

关键技术

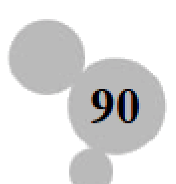
if 语句是最简单的条件判断语句，它对某段程序的执行附加一个条件，如果条件成立，就执行这段程序；否则跳过这段程序，执行下面的程序。

本实例的关键点是 if 语句的灵活运用。当 expr 的值为 TRUE 时，执行 statement1 语句。

```
if(expr){statement1;}
```

参数说明：

expr：条件判断语句。



statement：符合条件的执行代码。

设计过程

创建 index.php 文件，当单击“评定”按钮时，程序自动利用 POST 方法接收数据，并且利用 if 语句对接收的数据进行判断，再用 echo 语句输出结果，其代码如下：

```
<?php
    if($_POST[sub]){                                        //通过 POST 方式传递参数
        if($_POST[text] == 100){                            //当成绩为 100 时
            echo "成绩优秀";                                 //输出
        }
        if($_POST[text] >= 60 and $_POST[text] < 100){      //当成绩大于等于 60 小于 100 时
            echo "成绩良好";                                 //输出
        }
        if($_POST[text] < 60){                              //当成绩小于 60 时
            echo "不及格";                                   //输出
        }
    }
?>
```

秘笈心法

心法领悟 059：if 语句的变体。

if 语句有多种变体，因此其应用范围非常广泛，在后面的很多实例中都可以找到它的身影。

实例 060　控制登录用户权限　高级

光盘位置：光盘\MR\02\060　趣味指数：★★★★

实例说明

用户在浏览一些网站时，网站本身会将用户定义为 VIP 会员或者游客，会员可以享有更多的权限，而游客对网站的浏览有权限限制，下面就来实现这一功能。本实例通过 if…else…语句实现对登录用户权限的控制，运行结果如图 2.31 所示。

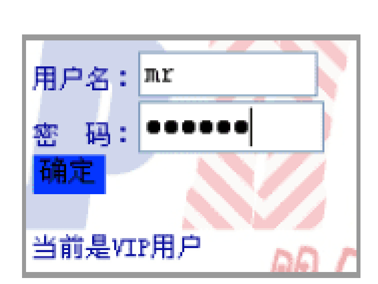


图 2.31　控制登录用户权限

关键技术

大多时候，总是需要在满足某个条件时执行一条语句，不满足时执行另一条语句。为了适应这样的需求，用户可以使用 if 语句的 else 子句，其语法如下：

```
if(expr){statement1;}else{statement2;}
```

如果表达式 expr 的值为 TRUE，则执行 statement1 语句；如果表达式 expr 的值为 FALSE，则执行 statement2 语句。

设计过程

（1）创建 PHP 脚本文件，当单击“确定”按钮时，程序通过 POST 方法接收传递过来的数据，并且通过 if…else…选择语句设定条件，当符合条件时利用 echo 语句输出“当前是 VIP 用户”，否则输出“您只是一个游

客”，其代码如下：

```
<?php
    if($_POST['sub']){
        //判断通过 POST 接收的参数结果是否符合条件
        if($_POST['user'] == 'mr' and $_POST['pwd'] == 'mrsoft'){
            echo "当前是 VIP 用户";                          //符合条件的结果
        }else{
            echo "您只是一个游客";                           //不符合条件的结果
        }
    }
?>
```

（2）将该文件存储于\MR\02\060 文件夹中，命名为 index.php。运行结果如图 2.31 所示。

秘笈心法

心法领悟 060：使用 if…else…语句的优点。

if…else…语句是 if 语句的另一种形式，结构上相对 if 语句要清晰。

实例 061　网页框架的制作

光盘位置：光盘\MR\02\061

高级

趣味指数：★★★★

实例说明

类似于留言板等一些功能较单一的网站，可以通过 switch 语句制作网页框架将所有内容包含到主页中。本实例通过 switch 语句、include 语句来演示网页框架的制作，运行结果如图 2.32 所示。



图 2.32　网页框架的制作

关键技术

switch 语句和 break 语句的灵活运用是本实例的关键。

（1）switch 语句和具有同样表达式的一系列的 if 语句相似。很多场合下需要把同一个变量（或表达式）与很多不同的值比较，并根据它等于哪个值来执行不同的代码，其语句格式如下：

```
switch(variable){
case value1:
        statement1;
        break;
case value2:
```



```
...
default:
        default statement n;
}
```

switch 语句根据 variable 的值，依次与 case 中的 value 值相比较，如果不相等，继续查找下一个 case；如果相等，就执行对应的语句，直到 switch 语句结束或遇到 break 为止。一般 switch 语句最终都有一个默认值 default，如果在前面的 case 中没有找到相符的条件，则输出默认语句（与 else 语句类似）。

（2）break 语句用于结束当前 for、foreach、while、do...while 或者 switch 结构的执行。

设计过程

（1）创建 PHP 脚本文件，首先，利用 switch 语句获取从地址栏传递的数据，并且将数据与 switch 语句中的 case 做比较，当地址栏传递的数据与 case 中的数据相等时，包含指定文件，并且利用 break 语句跳出 switch 语句，其代码如下：

```
<?php
        switch($_GET['link']){                              //通过地址栏接收参数
                case "主页":                                //如果值为主页
                include('inc.php');                         //包含 inc.php 文件
                break;                                      //结束执行 switch 语句
                case "注册":                                //如果值为注册
                include('inc_1.php');
                break;
                case "浏览":                                //如果值为浏览
                include('inc_2.php');
                break;
                default:                                    //默认情况下，包含 inc.php 文件
                include('inc.php');
        }
?>
```

（2）将该文件存储于\MR\02\061 文件夹下，命名为 index.php。运行结果如图 2.32 所示。

秘笈心法

心法领悟 061：switch 语句的实现原理。

switch 语句在执行时，即使遇到符合要求的 case 语句段，也会继续往下执行，直到 switch 结束。为了避免这种浪费时间和资源的行为，一定要在每个 case 语句段后添加 break 跳转语句跳出当前循环。

实例 062	图片验证码 光盘位置：光盘\MR\02\062	高级 趣味指数：★★★★

实例说明

验证码的作用是防止用户恶意注册而降低数据库和网站本身的性能。本实例应用 switch 语句通过 PHP 图像函数来演示图片验证码的制作，运行结果如图 2.33 所示。



图 2.33　图片验证码

关键技术

本实例的关键点是验证码函数和 switch 语句的联合应用，其应用到的函数如下：

（1）header()函数

当参数为 Content-type:image/png 时使用 header()函数表示将图片输出到浏览器。

```
void header ( string string [, bool replace [, int http_response_code]]);
```

（2）imagecreatetruecolor()函数

新建一个真彩色图像，imagecreatetruecolor()函数返回一个图像标识符，代表了一幅大小为 x_size 和 y_size 的黑色图像。

```
resource imagecreatetruecolor ( int x_size, int y_size);
```

（3）imagecolorallocate()函数

为一幅图像分配颜色，imagecolorallocate()函数返回一个标识符，代表了由给定的 RGB 成分组成的颜色。image 参数是 imagecreate() 函数的返回值；red、green 和 blue 分别是所需要的颜色的红、绿、蓝成分。这些参数是 0~255 的整数或者十六进制的 0x00~0xFF。imagecolorallocate()函数必须被调用以创建每一种用在 image 所代表的图像中的颜色。

```
int imagecolorallocate ( resource image, int red, int green, int blue);
```

（4）imagettftext()函数

用 TrueType 字体向图像写入文本，imagettftext()函数是将字符串 text 添加到 image 所代表的图像上，从坐标(x,y)（左上角为(0, 0)）开始，角度为 angle，颜色为 color，使用 fontfile 所指定的 TrueType 字体文件。根据 PHP 所使用的 GD 库的不同，如果 fontfile 没有以 '/'开头，则 '.ttf' 将被加到文件名之后并且会搜索库定义字体路径。

```
array imagettftext ( resource image, int size, int angle, int x, int y, int color, string fontfile, string text);
```

（5）imagegif()函数

以 GIF 格式将图像输出到浏览器或文件，imagegif()函数从 image 图像以 filename 为文件名创建一个 GIF 图像。image 参数是 imagecreate() 函数的返回值。

```
int imagegif ( resource image [, string filename]);
```

（6）int imagestring()函数

水平地画一行字符串，int imagestring()函数用 col 颜色将字符串 s 画到 image 所代表的图像的(x,y)坐标处（图像的左上角为(0, 0)）。如果 font 是 1、2、3、4 或 5，则使用内置字体。

```
int imagestring ( resource image, int font, int x, int y, string s, int col);
```

设计过程

（1）创建 index.php 文件，通过 GET 方式接收由地址栏传递的数据并且将参数与 case 的值进行比较，当两者相等时引入指定图片，通过 switch 语句进行操作，其代码如下：

```
<?php
	switch($_GET['link']){                          //取得通过地址栏传递的参数
		case "文字":                            //当参数等于文字时
?>
		<img src="yzm_1.php">                   //引入图片
<?php
		break;                                  //结束执行 switch 语句
		case "数字":                            //当参数等于数字时
?>
		<img src="yzm_2.php">                   //引入图片
<?php
		break;                                  //结束执行 switch 语句
		default:                                //默认情况下
?>
		<img src="yzm_2.php">
<?php
}
?>
```



（2）创建 yzm_1.php 文件，应用图像处理函数生成文字验证码，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
header("Content-type:image/gif");                                  //设置页面类型
$image = imagecreatetruecolor(80,30);                              //创建画布
$font = 'Font/FZHCJW.TTF';                                         //定义字体
$bg = imagecolorallocate($image,255,255,255);                      //定义背景颜色
$color = imagecolorallocate($image,255,0,255);
imagettftext($image,20,0,8,20,$color,$font,"mrsoft");              //输出文字
imagegif($image);                                                  //生成图像
?>
```

（3）创建 yzm_2.php 文件，应用图像处理函数生成数字验证码，其代码如下：

```
<?php
header("Content-type:image/gif");                                  //设置页面格式
$image = imagecreatetruecolor(40,20);                              //生成画布
$bg = imagecolorallocate($image,225,225,0);                        //定义背景颜色
$color = imagecolorallocate($image,255,0,255);                     //定义图像颜色
for($b = 0;$b <4; $b ++){                                          //生成随机验证码
        $a .= rand(1,9);
}
imagestring($image,3,8,5,$a,$color);                               //输出验证码
imagegif($image);                                                  //生成图像
?>
```

秘笈心法

心法领悟 062：开启 GD2 支持。

PHP 5.0 中 GD2 函数库已经作为扩展被默认安装，但目前有些版本中，还需要对 php.ini 文件进行设置来激活 GD2 函数库。用文本编辑工具，如记事本等程序打开 php.ini 文件，将文件中的“;extension=php_gd2.dll”选项中的分号“;”删除，保存修改后的文件，并重新启动 Apache 服务器即可激活 GD2 函数库。

实例 063	健康生活提醒 光盘位置：光盘\MR\02\063	高级 趣味指数：★★★★

实例说明

一些手机在开机时会出现开机问候语或者今日日程提示。本实例就是设计类似这样的程序，通过 switch 语句根据当前日期，给出健康生活提示信息，运行结果如图 2.34 所示。



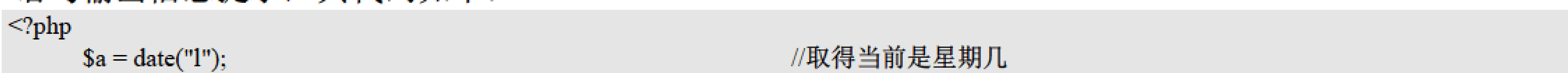
图 2.34　健康生活提醒

关键技术

本实例的关键点是利用 switch 语句，将地址栏传递的数据与 switch 语句中的 case 值进行比较，符合条件的利用 echo 语句输出对应数据。

设计过程

（1）创建 PHP 脚本文件。首先通过 date()函数取得当前是星期几，然后利用 switch 语句进行操作，并通过 echo 语句输出信息提示，其代码如下：



```
<?php
        $a = date("l");                                            //取得当前是星期几
```

```
        switch($a){
            case "Monday":                                          //当前如果是星期一
            echo "今天是星期一，一周忙碌生活开始了";                //输出
            break;
            case "Tuesday":                                         //当前如果是星期二
            echo "今天是星期二，电视台下午两点以后部分休息";        //输出
            break;
            case "Wednesday":                                       //当前如果是星期三
            echo "今天是星期三，下午有乒乓球比赛";                  //输出
            break;
            case "Thursday":                                        //当前如果是星期四
            echo "今天是星期四，晚上有 NBA 的重播";                 //输出
            break;
            case "Friday":                                          //当前如果是星期五
            echo "今天是黑色星期五。。。。。。";                    //输出
            break;
            case "Saturday":                                        //当前如果是星期六
            echo "今天是星期六，明天就放假了哈哈哈";                //输出
            break;
            case "Sunday":                                          //当前如果是星期日
            echo "今天是星期天，可以玩上一整天":                    //输出
        }
?>
```

（2）将该文件存储于\MR\02\063 文件夹中，命名为 index.php。运行结果如图 2.34 所示。

秘笈心法

心法领悟 063：比较 switch 语句与 if…else…语句。

switch 语句与 if…else…语句在功能上基本相同，但是 switch 语句更加灵活，格式更加鲜明，结构更加清晰。

2.5 循环语句

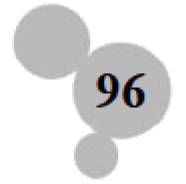
在实际应用中，经常会遇到一些操作并不复杂，但需要反复处理的问题，为此，PHP 提供了循环语句来实现循环结构的程序设计。循环语句是指能够按照一定的条件重复执行某段功能代码的代码结构。循环语句分为以下 4 种：

- ☑ while 循环语句。
- ☑ do...while 循环语句。
- ☑ for 循环语句。
- ☑ foreach 循环语句。

实例 064	员工生日列表 光盘位置：光盘\MR\02\064	高级 趣味指数：★★★★

实例说明

while 循环结构的作用是重复执行一段代码或完成相同的动作。本实例通过 while 语句循环输出数组中存储的员工生日信息，运行结果如图 2.35 所示。



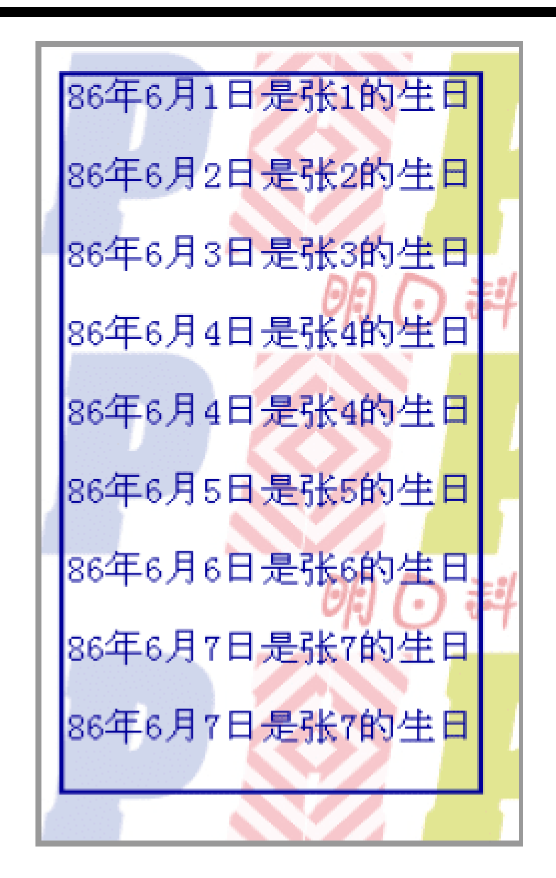


图 2.35　员工生日列表

关键技术

while 循环是 PHP 中最简单的循环语句。while 循环语句根据某一条件进行判断，决定是否执行循环，其语法格式是：

```
while (expr){
statement
}
```

当表达式 expr 的值为 TRUE 时，将执行 statement 语句，执行结束后，再返回到 expr 表达式继续进行判断。直到表达式的值为 FALSE 才跳出循环，执行大括号后面的语句。

设计过程

创建 index.php 文件。首先，利用数组函数将数据信息存储在数组变量中。然后，定义循环变量的初始值为 0，最大值不超过数组长度，并在循环内部做自增运算。最后，利用 while 循环输出结果，其代码如下：

```
<?php
	$a = array("86 年 6 月 1 日是张 1 的生日","86 年 6 月 2 日是张 2 的生日","86 年 6 月 3 日是张 3 的生日","86 年 6 月 4 日是张 4 的生日","86 年 6 月 4 日是张 4 的生日","86 年 6 月 5 日是张 5 的生日","86 年 6 月 6 日是张 6 的生日","86 年 6 月 7 日是张 7 的生日","86 年 6 月 7 日是张 7 的生日");		//定义数组
$b = 0;								//给变量赋初始值
	while($b < count($a)){					//根据数组长度循环输出
		echo $a[$b]."<br><br>";				//输出数组元素
		$b++;
	}
?>
```

秘笈心法

心法领悟 064：介绍 while 语句的变体。

while 循环语句还有另一种表示形式，即 do...while，详细说明请参见实例 065。

实例 065	员工详细信息浏览 光盘位置：光盘\MR\02\065	高级 趣味指数：★★★★

实例说明

本实例通过 do…while 循环语句循环读取数据库中的数据，输出员工的详细信息，运行结果如图 2.36 所示。

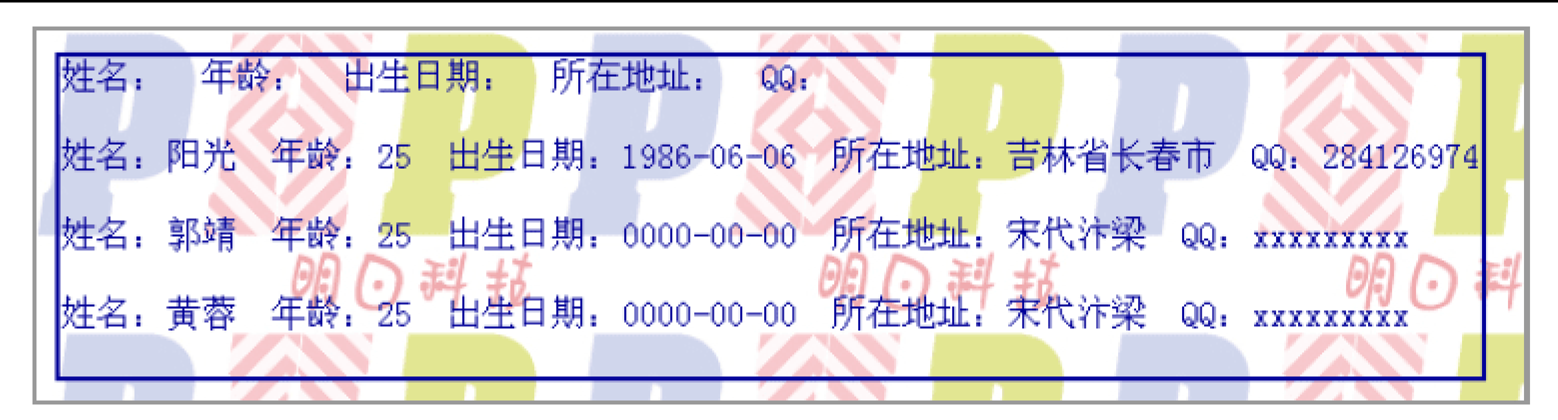


图 2.36　员工详细信息浏览

关键技术

while 语句还有另外一种表示形式，即 do…while。do…while 循环语句和 while 循环语句非常类似，只是 do…while 循环语句在循环底部检测循环表达式，而不是在循环的顶部进行检测。do…while 循环语句的语法格式如下：

```
do{
        statement
}while(expr);
```

设计过程

（1）创建 PHP 脚本文件。首先，利用 mysql_connect()函数连接 MySQL 数据库，并通过 mysql_query()函数定义页面的编码格式。然后，定义 SQL 查询语句并返回结果集。最后，通过 do…while 语句循环输出数据库中的数据，其代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111")                //连接数据库服务器
mysql_select_db("db_database02",$conn);                        //连接数据库
mysql_query("SET NAMES GBK");                                  //设置编码格式
$rs = mysql_query("select * from tb_while");                   //执行查询
do{
        echo "姓名：".iconv('gbk','utf-8',$rst[1])."  年龄：".iconv('gbk','utf-8',$rst[2])."  出生日期：
".iconv('gbk','utf-8',$rst[3])."  所在地址：".iconv('gbk','utf-8',$rst[4])."  QQ：".iconv('gbk','utf-8',$rst[5])."<br><br>";
}while($rst = mysql_fetch_array($rs));
?>
```

（2）将该文件存储于\MR\02\065 文件夹中，命名为 index.php。运行结果如图 2.36 所示。

秘笈心法

心法领悟 065：使用 do…while 语句的注意事项。

在使用 do…while 语句之前，要考虑程序是否有必要在判断条件之前运行一次，如果没有必要，尽量不要使用该语句，否则可能会出现意外的输出结果。

实例 066　员工信息的批量删除

光盘位置：光盘\MR\02\066

高级　趣味指数：★★★★

实例说明

在操作数据库的过程中可能会出现很多无用的冗余信息，可以使用 while 循环语句将其循环删除。本实例通过 while 循环语句实现员工信息的批量删除，运行结果如图 2.37 所示。

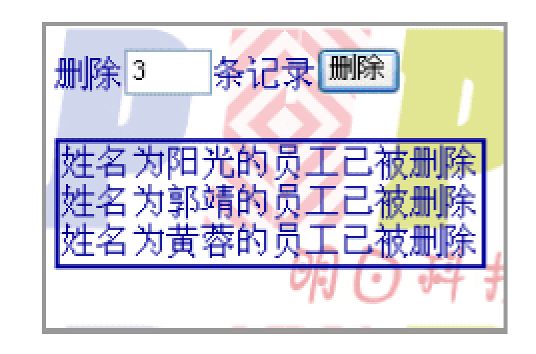
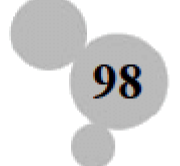


图 2.37　员工信息的批量删除



关键技术

本实例的关键点是 while 循环语句的灵活运用。根据提交的参数值，执行 while 循环语句，在循环体中执行删除操作，完成数据的循环删除操作。

设计过程

（1）创建 PHP 脚本文件。首先，连接 MySQL 数据库服务器，选择 MySQL 中名称为 db_while 的数据库，并设置页面的编码格式。然后，定义循环变量$a 和$b，将第一层 while 循环中的循环条件设置为$b<$a，然后，再次利用 while 语句循环输出数据表中的数据，并执行数据的删除操作，其代码如下：

```
<?php
if($_POST[sub]){                                                    //通过 POST 方式获取参数
        $conn = mysql_connect("localhost","root","111");            //连接 MySQL 数据库
        mysql_select_db("db_while",$conn);                          //连接数据库
        mysql_query("SET NAMES GBK");                               //定义编码格式
        $a = $_POST[te];                                            //接收文本框参数
        $b = 0;                                                     //定义变量
            while($b < $a){                                         //while 循环
                $rs = mysql_query("select * from tb_while");        //执行查询操作
                while($rst = mysql_fetch_array($rs)){               //将查询结果保存在数组中
                    $sql = "delete * from tb_while where id = $b";  //SQL 语句
                    mysql_query($sql);                              //执行删除操作
                    echo "姓名为".$rst[name]."的员工已被删除<br>";  //输出被删除的员工姓名
                    $b++;
                }
            }
    }
?>
```

（2）将该文件存储于\MR\02\066 文件夹下，命名为 index.php。运行结果如图 2.37 所示。

秘笈心法

心法领悟 066：使用 while 语句嵌套的注意事项。

while 循环语句允许语句嵌套，但在嵌套的过程中要注意保持层次结构的清晰。

实例 067	表格的动态创建 光盘位置：光盘\MR\02\067	高级 趣味指数：★★★★

实例说明

很多时候 PHP 编码需要和 HTML 编码混编，例如，输出表格打印 HTML 标签等。本实例应用两层 for 语句循环制作动态表格，运行结果如图 2.38 所示。

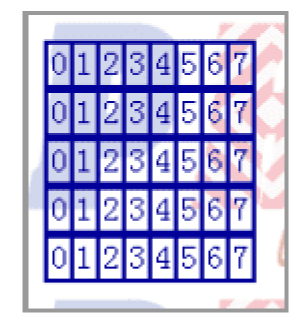


图 2.38　表格的动态创建

关键技术

for 循环语句能够按照已知的循环次数进行循环操作，主要应用于多条件情况下的循环操作，不适合在单一条件下使用。这一点从该语句的语法中就可以看出，其条件的表达式有 3 个，语法格式如下：

```
for (expr1; expr2; expr3){
statement;
}
```

其中，expr1 为变量初始赋值；expr2 为循环条件，即在每次循环开始前求值，如果其值为真，则执行 statement，否则，跳出循环，继续往下执行；expr3 为变量递增或递减，即每次循环后被执行。

设计过程

（1）创建 PHP 脚本文件。首先，定义 for 循环中的循环变量$a。设定$a 的初始值为 0，最大值为 4，并且每执行一次循环就做一次自增运算。然后，利用循环嵌套的知识定义内层循环，内层循环的循环变量$b 的初始值为 0，最大值为 7，并且同样做自增运算。最后，利用 echo 语句输出信息，其代码如下：

```
<?php
    for($a = 0;$a < 5;$a++){                                    //1 层 for 循环
        echo "<table border='1'bordercolor='#000099' bordercolorlight='#000099' cellspacing='0'>";
        echo "<tr>";                                            //表格标签
        for($b = 0;$b < 8;$b++){                                //2 层 for 循环
            echo "<td>";                                        //表格标签
            echo $b;                                            //输出数据
            echo "</td>";
        }
        echo "</tr>";
        echo "</table>";
    }
?>
```

（2）将该文件存储于\MR\02\067 文件夹下，命名为 index.php。运行结果如图 2.38 所示。

秘笈心法

心法领悟 067：for 循环语句。

for 循环语句在 PHP 循环结构中比较常用，结构也比较简单，所以一定要灵活掌握此循环语句。在本实例中，虽然其定义变量的最大值分别为 4 和 7，但是由于在 for 循环中，其条件表达式的值是从等于 0 开始，所以循环语句分别执行了 5 次和 8 次。

实例 068　SESSION 购物车中数据的读取

光盘位置：光盘\MR\02\068

高级

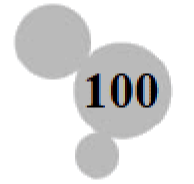
趣味指数：★★★★

实例说明

购物车是电子商务系统中的常见模块，SESSION 作为一种会话机制作用不可小觑。本实例综合运用 while 语句、switch 语句、if 语句和 for 语句实现一个简单的 SESSION 购物车功能，运行结果如图 2.39 所示。

1
山地自行车
1件
600元
30元（自理）
金牌用户VIP享受8.8折优惠！
2
液晶电脑显示器
1件
900元
30元（自理）
金牌用户VIP享受8.8折优惠！
3
液晶电脑显示器
1件
900元
30元（自理）
金牌用户VIP享受8.8折优惠！

图 2.39　SESSION 购物车中数据的读取



关键技术

（1）通过 switch 语句根据地址栏中传递的参数值，实现在不同商品页面之间的跳转。

（2）通过 if 语句对 form 表单提交的值进行判断，进而执行不同的操作。

（3）通过 while 语句和 for 语句完成购物车中购买商品的输出。

设计过程

（1）创建 index.php 文件，通过 switch 语句根据地址栏中传递的参数值，实现在不同商品页面之间的跳转，其代码如下：

```
<?php
	switch($_GET[link]){
		case "电脑":
		include  once("in  1.php");
		break;
		case "家庭影院":
		include_once("in_2.php");
		break;
		case "全自动洗衣机":
		include_once("in_3.php");
		break;
		case "数码摄像机":
		include_once("in_4.php");
		break;
		case "山地自行车":
		include_once("in_5.php");
		break;
		default:
		include_once("in_image.php");
	}
?>
```

（2）创建 config.php 文件。首先，编辑 form 表单，执行不同的操作。然后，应用 if 语句根据 form 表单提交的值进行判断，进而执行不同的操作。最后，通过 while 和 for 语句循环输出购物车中存储的商品，其关键代码如下：

```
	<form action="" method="post">
		我要购买<input type="text" size="2" value ="1" name="num">件
	      <input type="submit" name="sub" value="放入购物车" style="background-image:url(pic/button_2.jpg);color:red">
		<input type="submit" name="sub1" value="结账" style="background-image:url(pic/button_3.jpg);color:red">
		<input type="submit" name="sub2" value="首页" style="background-image:url(pic/button_1.jpg);color:red">
	</form>
<?php
		if($_POST[sub2]){
			echo "<script>location.href='index.php'</script>";
		}
		if($_POST[sub]){
			$str = explode("#",$_SESSION[id]);
			include("class/mysql.php");
			$sql = "insert into tb_shop(id,name,number,money,moneyother,other)values('','$str[0]','$str[1]','$str[2]','$str[3]','$str[4]')";
			if(mysql_query($sql)){
				mysql_close($conn);
				echo "<script>alert('已经放入购物车')</script>";
			}
		}
	if($_POST[sub1] || $_GET['sub1']){
		require_once("class/mysql.php");
		$sq = "select * from tb_shop";
		$rs = mysql  query($sq);
?>
		<form action ="config/table.php" method="post">
			<input type="submit" name="sub3" value="删除" style="background-image:url(pic/button_3.jpg);color:red">
```

```
        </form>
<?php
    while($rst = mysql_fetch_row($rs))
        for($a = 0;$a < 6;$a++){
?>
                <tr><td><?php echo $rst[$a]?></td></tr>
<?php
        }
?>
    </table>
    <a href="config/money.php"><img src="pic/bu.jpg"></a>
<?php
    }
?>
```

（3）创建 table.php 文件，执行删除指定商品的操作，其关键代码如下：

```
<?php
require_once("../class/mysql.php");                          //包含数据库连接文件
        $sq = "select * from tb_shop";                        //定义 SQL 查询语句
        $rs = mysql_query($sq);                               //执行查询操作
?>
    <table>
<?php
        while($rst = mysql_fetch_row($rs)){                   //循环输出商品信息
?>
        <tr><td>商品编码：<?php echo $rst[0];?></td><td>商品名称：<?php echo $rst[1];?></td></tr>
        <tr><td>购买数量：<?php echo $rst[2];?><hr></td><td>金额总计：<?php echo $rst[3];?><hr></td></tr>
        <!--tr><td>商品运费：<?php echo $rst[4];?><hr></td><td>其他声明：<?php echo $rst[5];?><hr></td></tr-->
<?php
        }
?>
    </td>
    </tr></table></td></tr></table>
<?php
if($_POST[sub4]){
    $sqlser = "delete from tb_shop where id = '$_POST[tt]'";  //定义 SQL 语句
    if(mysql_query($sqlser)){                                 //执行删除操作
        echo "<script>alert('商品删除成功')</script>";
        echo "<script>window.location.href='table.php';</script>";
    }
}
if($_POST[sub5]){
    echo "<script>window.location.href='../index.php?link=电脑&sub1=1';</script>";
}
?>
```

（4）编辑 mysql.php 文件，实现与数据库的连接。

秘笈心法

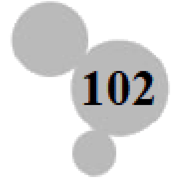
心法领悟 068：知识对接。

购物车的知识相对较复杂，在本书第 5 章的实例 266 中将对购物车进行详细讲解。

实例 069	员工信息的管理 光盘位置：光盘\MR\02\069	高级 趣味指数：★★★★

实例说明

员工信息一般存在于人事信息平台的后台系统中，对员工信息的管理本身可以作为一个典型的模块被使用。本实例通过 for 循环语句实现对员工数据的读取、修改和删除操作，运行结果如图 2.40~图 2.42 所示。



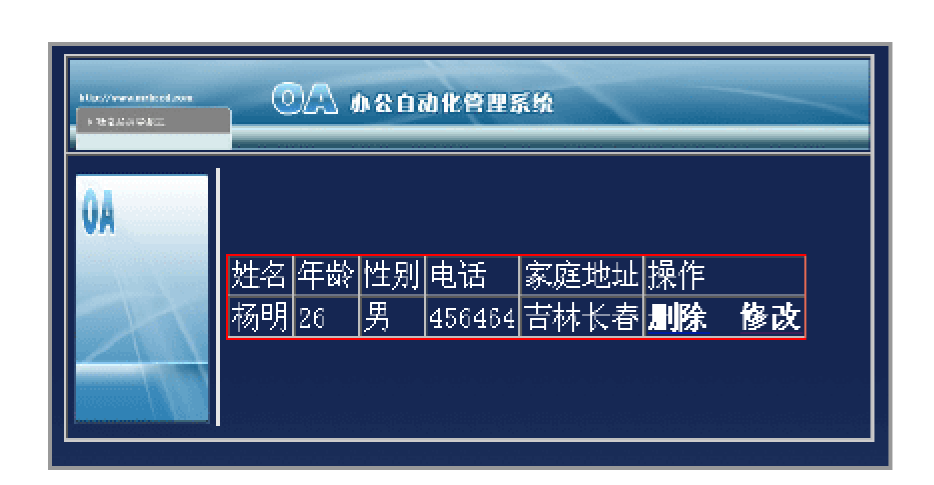



图 2.40　员工数据的读取



图 2.41　员工数据的删除

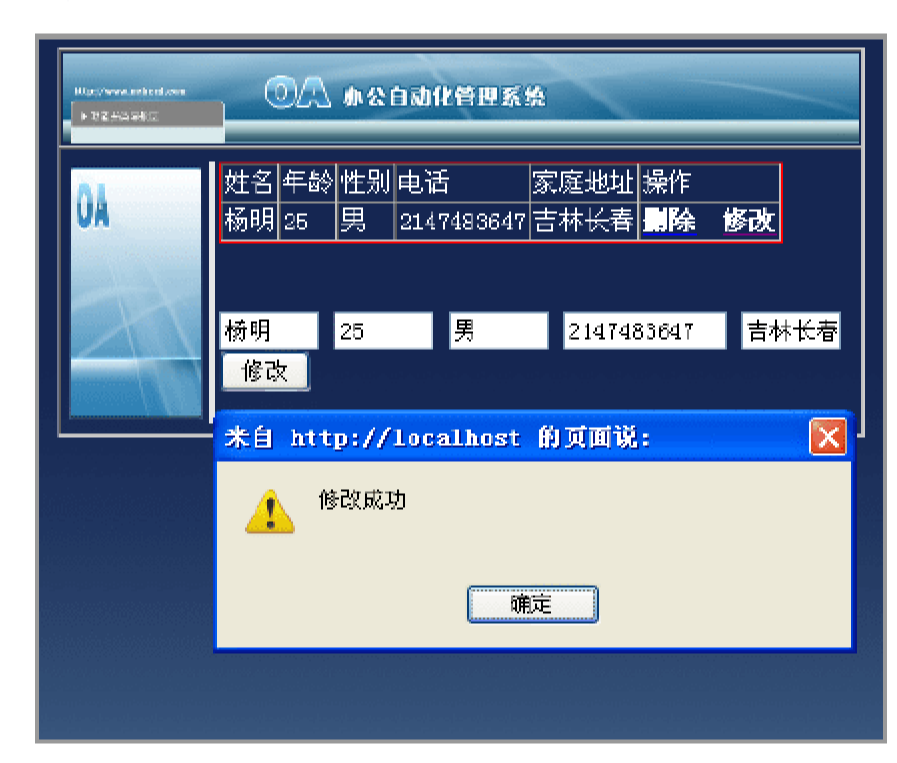


图 2.42　员工数据的修改

关键技术

本实例通过 for 循环语句读取查询结果集中的数据，完成数据的循环输出，并且创建删除和修改超链接，应用 if 语句对超链接传递的参数值进行判断，实现对数据的删除和修改操作。有关 for 循环语句的详细说明请参见实例 067，if 语句的详细讲解请参见实例 059。

设计过程

创建 index.php 文件。首先，连接 MySQL 数据库执行查询操作取得查询结果集。然后，通过 for 语句循环输出结果集中的数据。最后，通过 if 语句对超链接的参数值进行判断，进而执行删除和修改操作，其关键代码如下：

```
<?php
    $link = mysql_pconnect("localhost", "root", "111");                    //连接 MySQL
    mysql_select_db("db_database02",$link);                                //连接数据库
     mysql_query("SET NAMES GBK");                                         //设置编码格式
    $query = "SELECT * FROM tb_for_1";                                     //SQL 语句
    $result = mysql_query($query);                                         //执行查询操作
    for ($i = mysql_num_rows($result) - 1; $i >= 0; $i--) {                //for 循环
        if (!mysql_data_seek($result, $i)) {                               //如果指针不向下移动
            echo "Cannot seek to row $i: " . mysql_error() . "\n";
            continue;
        }

        if(!($row = mysql_fetch_object($result)))                          //如果结果集不存在
            continue;

        echo "<tr><td>$row->name</td><td>$row->age</td><td>$row->sex</td><td>$row->tel</td><td>$row->address</td>
      <td><a href='1.php?id=$row->id&delete=1'><b>删除</b></a>  <a href='1.php?id=$row->id&update=1'><b>修改
</b></a></td></tr>";
```

```
    }
    mysql_free_result($result);                                           //释放结果集
if($_GET[delete] == 1){                                                   //如果地址栏参数 delete 等于 1
            $sql = "delete from tb_for_1 where id = '$_GET[id]'";         //SQL 语句
            if(mysql_query($sql)){                                        //执行删除操作
                echo "<script>alert('删除成功');location.href='1.php'</script>";   //提示框信息
            }
}
if($_GET[update] == 1){                                                   //如果地址栏参数 update 等于 1
            $sql = "select * from tb_for_1 where id ='$_GET[id]'";        //SQL 语句
            $rs = mysql_query($sql);                                      //执行查询语句
            $rst = mysql_fetch_array($rs);                                //取得结果集
?>
</table><br><br>
        <form action="" method="post">                                    //表单
          //使文本框初始值为数据库中结果
          <input name="name" type="text" value="<?php echo $rst[name];?>" size="6">
          <input name="age" type="text" value="<?php echo $rst[age];?>" size="6">
          <input name="sex" type="text" value="<?php echo $rst[sex];?>" size="6">
          <input name="tel" type="text" value="<?php echo $rst[tel];?>" size="11">
          <input name="address" type="text" value="<?php echo $rst[address];?>" size="6">
          <input type="submit" name="sub1" value="修改">
         </form></td></tr></table></td></tr></table>
<?php
            if($_POST[sub1]){                                             //如果按钮被单击
                $sql = "update tb_for_1 set name = '$_POST[name]',age = '$_POST[age]',sex = '$_POST[sex]',tel = '$_POST[tel]',address =
'$_POST[address]' where id =  '$_GET[id]'";//sql 语句

                if(mysql_query($sql)){                                    //执行更新操作
                    echo "<script>alert('修改成功');location.href='1.php'</script>";    //提示框信息
                }
            }
}
?>
```

秘笈心法

心法领悟 069：使用地址栏传递参数存在风险。

通过地址栏传递参数是存在风险的，一般情况下应使用 POST 方式传递参数。如果一定要使用 GET 方式传递参数，最好应用 urlencode()函数对参数进行编码。

实例 070　网页版九九乘法表

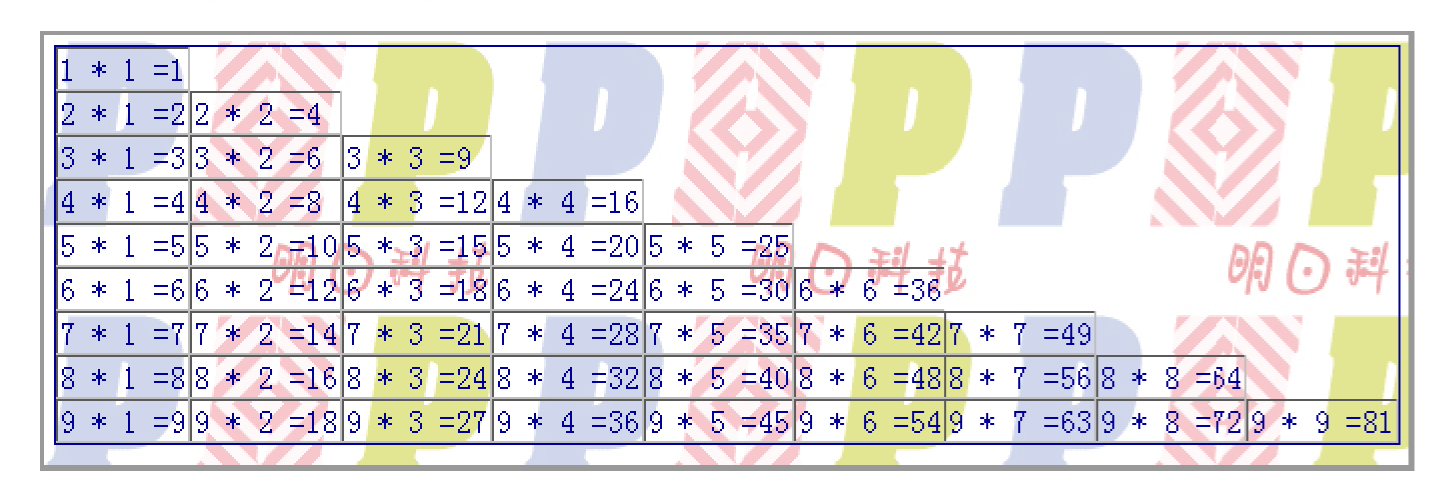
光盘位置：光盘\MR\02\070

高级

趣味指数：★★★★

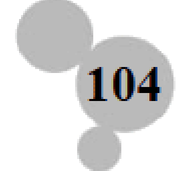
实例说明

在编程世界里，算法是一门独立于语法、函数之外的知识，它着重考验用户的逻辑思维能力并且与数学知识息息相关。本实例通过 for 循环语句实现网页版九九乘法表，运行结果如图 2.43 所示。



1 * 1 =1								
2 * 1 =2	2 * 2 =4							
3 * 1 =3	3 * 2 =6	3 * 3 =9						
4 * 1 =4	4 * 2 =8	4 * 3 =12	4 * 4 =16					
5 * 1 =5	5 * 2 =10	5 * 3 =15	5 * 4 =20	5 * 5 =25				
6 * 1 =6	6 * 2 =12	6 * 3 =18	6 * 4 =24	6 * 5 =30	6 * 6 =36			
7 * 1 =7	7 * 2 =14	7 * 3 =21	7 * 4 =28	7 * 5 =35	7 * 6 =42	7 * 7 =49		
8 * 1 =8	8 * 2 =16	8 * 3 =24	8 * 4 =32	8 * 5 =40	8 * 6 =48	8 * 7 =56	8 * 8 =64	
9 * 1 =9	9 * 2 =18	9 * 3 =27	9 * 4 =36	9 * 5 =45	9 * 6 =54	9 * 7 =63	9 * 8 =72	9 * 9 =81

图 2.43　网页版九九乘法表



关键技术

本实例的关键点是利用 for 循环中的第 2 层循环控制第 1 层循环，以实现指定数据的输出。

设计过程

（1）创建 PHP 脚本文件，定义第 1 层循环中的循环变量为$a，设置$a 的初始值为 1，最大值为 9，并且每执行一次循环就做一次自增运算。然后，定义内层循环中的循环变量为$b，设置$b 的初始值为 1，最大值小于$a，并且做自增运算。最后，将第 2 层循环控制的数据输出到页面，其代码如下：

```
<?php
    for($a = 1;$a < 10;$a++){                         //第 1 层循环
        echo "<tr>";
        for($b = 1;$b <= $a;$b++){                    //第 2 层循环控制第 1 层循环输出
            echo "<td>";
            echo "$a * $b =".$a*$b;                   //输出结果
            echo"</td>";
        }
        echo "</tr>";
    }
?>
```

（2）将该文件存储于\MR\02\070 文件夹中，命名为 index.php。运行结果如图 2.43 所示。

秘笈心法

心法领悟 070：形成良好的算法思维。

算法不是一朝一夕就可以掌握的，是一个慢慢了解、形成思想的过程。

实例 071	读取数组购物车中的数据 光盘位置：光盘\MR\02\071	高级 趣味指数：★★★★

实例说明

用户在进行开发时，没有必要将小型的数据与数据库进行交互，可以直接将其存入数组，这样不仅可以节约开发时间，还可节省服务器资源。本实例通过对数组函数的相关操作实现读取数组购物车中的数据，运行结果如图 2.44 所示。



图 2.44　实现读取

关键技术

本实例的关键点是运用 foreach 语句遍历数组，取得 key 值和对应的 value 值。foeach 循环在 PHP 4.0 中擅长处理数组，是遍历数组的一种简单方法；在 PHP 5.0 中，新增加了对对象的支持。该语句的语法格式如下：

```
foreach (array  expression as $value)
statement
```

或

```
foreach (array_expression as $key => $value)
    statement
```

foreach 语句将遍历数组 array_expression，每次循环时，将当前数组中的值赋给$value（或是$key 和$value），同时，数组指针向后移动直到遍历结束。当使用 foreach 语句时，数组指针将自动被重置，所以不需要手动设置指针位置。

设计过程

（1）创建 PHP 脚本文件。首先，定义字符串变量，并且在变量中插入“#”作为字符串分割点。然后，利用 GET 方式接收地址栏传递的参数。当$_GET[push]的值等于 1 时，输出 JavaScript 成功提示。当$_GET[pop]的值等于 1 时，利用字符串拆分函数将字符串变量以“#”分割，并分别保存到数组中。最后，利用 foreach 语句将字符串数组中的 value 值输出到页面中，其代码如下：

```
<?php
    $str = "宝贝：液晶电脑显示器#一口价：900 元#运费：30 元（自理）";      //定义字符串
        if($_GET[push] == 1){                                               //通过 GET 方式取得参数
            echo "<script>alert('成功放入购物车');</script>";                //提示消息
        }
        if($_GET[pop] == 1){                                                //通过 GET 方式取得参数
            $strexplode = explode("#",$str);                                //分割字符串
            foreach($strexplode as $key => $value){                         //遍历数组
?>
            <tr><td><?php echo $value;?></td></tr>                          //输出结果
<?php
            }
        }
?>
```

（2）将该文件存储于\MR\02\071 文件夹中，命名为 index.php。运行结果如图 2.44 所示。

秘笈心法

心法领悟 071：对于 foreach 语句的说明。

foreach 语句仅能用于数组，当试图将其用于其他数据类型或者一个未初始化的变量时会产生错误。foreach 语句不支持用“@”来禁止错误信息。

实例 072	图像验证码的生成 光盘位置：光盘\MR\02\072	高级 趣味指数：★★★★

实例说明

用户注册后，系统为了保存用户的一些基本信息，包括用户名、密码等相关信息，以及跟踪和定位不同用户从而更好地服务于用户，同时为了提高站点的安全性，避免由于网速慢造成用户注册信息的重复提交，往往会在用户注册表中增加验证码功能。本实例通过 for 循环实现图像验证码的生成，运行结果如图 2.45 所示。

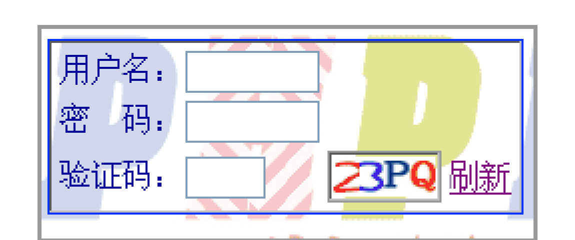
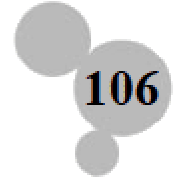


图 2.45　图像验证码的生成



关键技术

本实例的关键点是 for 循环语句和 rand()函数的灵活运用。通过 for 循环输出一个包含 4 个图像的验证码，验证码的取值由随机函数 rand()生成。

rand()函数可产生一个随机整数，其语法如下：

```
int rand ( [int min, int max] )
```

如果没有提供可选参数 min 和 max，rand()函数返回 0 到 RAND_MAX 之间的伪随机整数。例如，想得到 2~8（包括 2 和 8）之间的随机数，则写为 rand(2, 8)。

设计过程

（1）创建 PHP 脚本文件。首先，利用数组函数定义数组并分别将图片的地址信息存入数组中。然后，在 for 循环的循环体内，利用随机函数取得 0~3 中的随机数字并保存在变量$rand 中。最后，利用拼接字符串的方法输出图片，其代码如下：

```
<?php
	$array = array("pic/22.jpg","pic/3.jpg","pic/4.jpg","pic/5.jpg");		//定义数组
	for($a = 0;$a < 4;$a++){							//for 循环
		$rand = rand(0,3);							//0~3 的随机数字
		echo "<img src = '$array[$rand]'>";					//输出图像
	}
?>
```

（2）将该文件存储于\MR\02\072 文件夹中，命名为 index.php。运行结果如图 2.45 所示。

秘笈心法

心法领悟 072：图像验证码的刷新。

在本实例中，通过重新链接到首页来实现验证码的刷新操作。

2.6　跳 转 语 句

跳转控制语句包括 break 和 continue 两个。其中 break 语句可以终止当前的循环；而 continue 语句只能终止本次循环，并进入下一次循环中。

实例 073	控制页面中表情图的输出 光盘位置：光盘\MR\02\073	高级 趣味指数：★★★★

实例说明

break 语句的含义是结束当前 for、foreach、while、do...while 或者 switch 结构的执行。本实例通过 break 语句实现控制页面中表情图的输出，运行结果如图 2.46 所示。



图 2.46　控制页面中表情图的输出

关键技术

使用 break 关键字可以终止当前的循环，包括 while、do...while、for、foreach 和 switch 在内的所有控制语句，

其在 for 语句中的应用示例如下：

```
for($i=1;$i<=4;$i++){                          //应用 for 循环控制语句输出表情头像
if($i==4){                                     //判断变量是否等于 4
        break;                                 //如果等于 4，使用 break 语句跳转循环
}
```

本实例将 break 语句与 switch 语句联合使用，控制页面中表情图的输出。

设计过程

（1）创建 PHP 脚本文件，当单击“确定”按钮时，利用 switch 语句选择输出的图片并通过 break 语句跳出当前 switch 语句，其代码如下：

```
<?php
        if($_GET[sub]){                                    //通过 GET 方式接收参数
                switch($_GET['check']){                    //switch 语句
                        case "1":                          //如果值为 1
                        echo "<img src='image/1.gif'>";    //输出图像 1
                        break;                             //跳出语句
                        case "2":                          //如果值为 2
                        echo "<img src='image/2.gif'>";    //输出图像 2
                        break;                             //跳出语句
                        case "3":                          //如果值是 3
                        echo "<img src='image/3.gif'>";    //输出图像 3
                        break;                             //跳出语句
                        case "4":                          //如果值为 4
                        echo "<img src='image/4.gif'>";    //输出图像 4
                        break;                             //跳出循环
                }
        }
?>
```

（2）将该文件存储于\MR\02\073 文件夹中，命名为 index.php。运行结果如图 2.46 所示。

秘笈心法

心法领悟 073：break 语句的妙用。

break 语句可以接受一个可选的数字参数来决定跳出几重循环。

实例 074 控制页面中数据的输出数量

光盘位置：光盘\MR\02\074

高级

趣味指数：★★★★

实例说明

break 语句可以指定跳出循环的层数，一般被用在控制条件方面。本实例通过 break 语句实现控制页面中数据的输出数量，运行结果如图 2.47 所示。

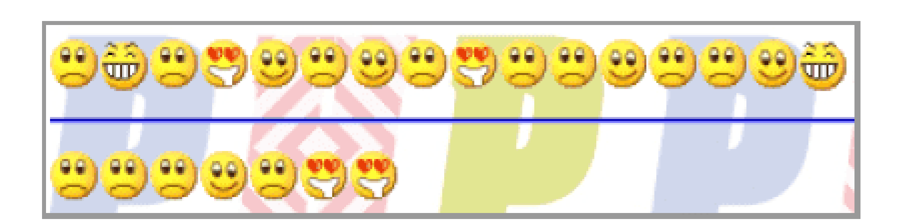
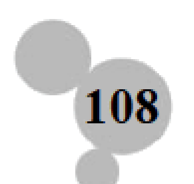


图 2.47　控制页面中数据的输出数量

关键技术

本实例运用 break 语句跳出指定的几重循环，其语法格式如下：

```
break n;
```



设计过程

（1）创建 PHP 脚本文件。首先，利用多重 for 循环嵌套输出全部图片。然后，在多重嵌套中定义 if 条件语句，当变量$rand 中的随机数值为 2 时，跳出 3 重循环，否则正常输出图片，其代码如下：

```
<?php
	for($a = 0;$a < 2;$a++){				//1 层循环
		for($b = 0;$b < 2;$b++){			//2 层循环
			for($c = 0;$c < 4;$c++){		//3 层循环
				$rand = rand(1,4);		//1~4 随机数字
				echo "<img src='image/$rand.gif>";	//输出图像
			}
		}
	}
	echo "<hr style='color:blue;'>";
	for($a = 0;$a < 2;$a++){				//1 层循环
		for($b = 0;$b < 2;$b++){			//2 层循环
			for($c = 0;$c < 4;$c++){		//3 层循环
				$rand = rand(1,4);		//1~4 随机数字
				if($rand != 2){			//条件语句
					echo "<img src='image/$rand.gif>";	//输出图像
				}else{
					break 3;		//结束 3 个层级的循环
				}
			}
		}
	}
?>
```

（2）将该文件存储于\MR\02\074 文件夹中，命名为 index.php。运行结果如图 2.47 所示。

秘笈心法

心法领悟 074：灵活运用跳转语句。

跳转语句一定要灵活运用。常用的跳转语句包括 break 语句和 continue 语句，continue 语句将在后面的实例中做详细介绍。

实例 075	动态改变页面中单元格的背景颜色 光盘位置：光盘\MR\02\075	高级 趣味指数：★★★★

实例说明

在循环结构中，有些数据是不需要的，想要跳过这些代码向下执行就需要用到 continue 语句。本实例通过 continue 语句实现动态改变页面中单元格的背景颜色，运行结果如图 2.48 所示。



图 2.48　动态改变页面中单元格的背景颜色

关键技术

在程序执行 break 后，程序将跳出循环执行流程开始继续执行循环体的后续语句。continue 跳转语句的作用没有 break 那么强大，continue 只能终止本次循环而进入下一次循环中。在执行 continue 语句后，程序将结束本

次循环的执行，并开始下一轮循环的执行操作。continue 跳转语句的流程控制如图 2.49 所示。

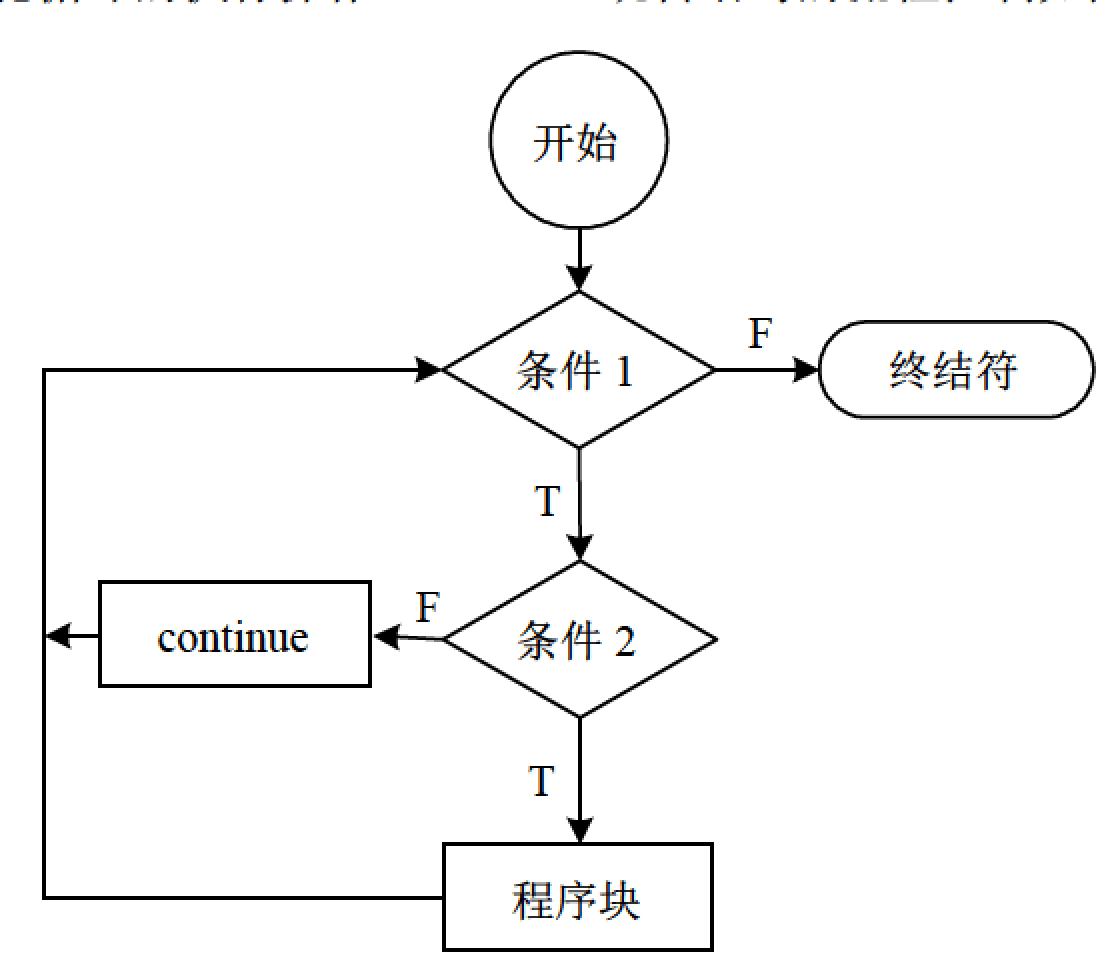


图 2.49　continue 流程控制

设计过程

（1）创建 PHP 脚本文件。首先，利用 for 循环定义循环变量的初始值为 0，最大值为 9，并且每执行完一次循环，循环变量做一次自增运算。然后，利用 if 语句定义条件，当循环变量小于等于 5 时，通过 dechex()函数获取 6 个十六进制随机数，否则终止本次循环。最后，将取得的 6 个十六进制数添加到<table>标签的 bgcolor 属性中，实现动态变色，其代码如下：

```
<?php
    for($b = 0;$b < 10;$b++){                    //循环语句
        if($b <= 5){                             //条件语句
            $rand.= dechex(rand(0,15));          //取得 6 个十六进制数
        }else{
            continue;                            //不满足条件，重新执行循环
        }
    }
?>
<table height="55" bgcolor="#<?php echo $rand;?>">//设置背景颜色
  <tr><td width="156"></td>
</tr></table>
```

（2）将该文件存储于\MR\02\075 文件夹中，命名为 index.php。运行结果如图 2.48 所示。

秘笈心法

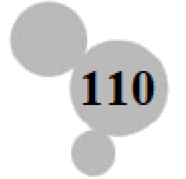
心法领悟 075：continue 语句的妙用。

continue 语句也可以指定跳转层级数。

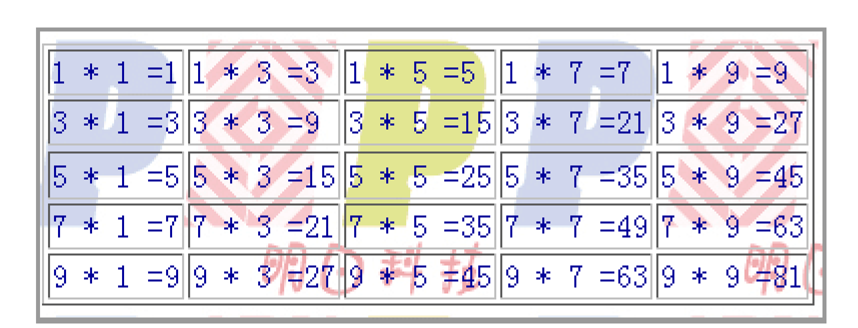
实例 076	屏蔽偶数次的数据输出 光盘位置：光盘\MR\02\076	高级 趣味指数：★★★★

实例说明

数学中有这样一道题：取得 0~10 中所有偶数或者奇数的和。如果想把这道题用 PHP 代码表示出来，则需



要使用跳转语句 continue 来完成。本实例通过跳转语句 continue 实现屏蔽偶数次的数据输出，运行结果如图 2.50 所示。



1 * 1 =1	1 * 3 =3	1 * 5 =5	1 * 7 =7	1 * 9 =9
3 * 1 =3	3 * 3 =9	3 * 5 =15	3 * 7 =21	3 * 9 =27
5 * 1 =5	5 * 3 =15	5 * 5 =25	5 * 7 =35	5 * 9 =45
7 * 1 =7	7 * 3 =21	7 * 5 =35	7 * 7 =49	7 * 9 =63
9 * 1 =9	9 * 3 =27	9 * 5 =45	9 * 7 =63	9 * 9 =81

图 2.50　屏蔽偶数次的数据输出

关键技术

本实例的关键点是 continue 语句的灵活运用，详细说明请参见实例 075。

设计过程

创建 PHP 脚本文件。首先通过 for 循环语句动态输出 HTML 的<tr>标签和数据，并且当循环变量为偶数时终止本次循环。然后，利用相同的方法动态输出 HTML 的<td>标签和数据。最后，在页面中输出如图 2.50 所示的表格，其代码如下：

```
<?php
	for($a = 1;$a < 10;$a++){				//1 层循环
		if($a % 2 == 0){				//如果为偶数则回到本循环开始位置
			continue;
		}
		echo "<tr>";
		for($b = 1;$b < 10;$b++){			//2 层循环
			if($b % 2 != 0){			//如果为偶数
				echo "<td>";
				echo "$a * $b =".$a*$b;	//输出结果
				echo "</td>";
			}else{
				continue;			//回到本循环的开始位置
			}
		}
		echo "</tr>";
	}
?>
```

秘笈心法

心法领悟 076：求余运算符。

利用“%”运算符可取得两个数的余数，本实例通过该运算符判断是否为偶数，如果想要判断是否为奇数，只需将条件设定为不等于 1 即可。

实例 077　跳过数据输出中指定的记录

光盘位置：光盘\MR\02\077

高级

趣味指数：★★★★

实例说明

本实例通过 continue 语句实现跳过数据输出中指定的记录，运行结果如图 2.51 所示。

关键技术

本实例应用 continue 语句跳过指定的循环，输出指定循环以外的数据。



图 2.51　跳过数据输出中指定的记录

设计过程

（1）创建 PHP 脚本文件。首先定义一个数组存储数据，然后定义下拉列表，其中下拉列表的值由 PHP 代码动态添加，当输入跳过记录的编号并单击“跳过”按钮时，利用 continue 语句，输出不包含指定数据的其他数据，其代码如下：

```
<?php
        //定义数组
        $array = array("PHP 典型模块","PHP 开发实战宝典","Java 开发实战宝典","PHP 网络编程自学手册");
?>
//表单
<form action="" method="post">
<textarea name="te" cols="20" rows="6">0、<?php echo $array[0];?>        1、<?php echo $array[1];?>        2、<?php echo $array[2];?>
3、<?php echo $array[3];?>
</textarea><br>
<input type="text" name="text" value="输入要跳过记录的编号"size="20" onFocus="this.value=''">
<input type="submit" name="sub1" value="跳过">
</form>
<?php
        if($_POST[sub1]){                                   //通过 POST 方式传递参数
                for($a = 0;$a < count($array);$a++){        //循环数组
                        if($a == $_POST[text]){             //当$a 等于文本框中的数据时回到循环开始位置
                                continue;
                        }else{
                                echo ' 《';
                                echo $array[$a];            //输出数组元素
                                echo '》 ';
                        }
                }
        }
?>
```

（2）将该文件存储于\MR\02\077 文件夹中，命名为 index.php。运行结果如图 2.51 所示。

秘笈心法

心法领悟 077：在 HTML 中插入 PHP 代码。

PHP 代码可以插入到 HTML 页面的任何位置，也可以插入到 HTML 的任何一个可行性标签中。

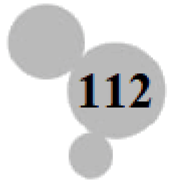
实例 078　执行指定次数的循环

光盘位置：光盘\MR\02\078

高级　趣味指数：★★★★

实例说明

本实例利用 break 语句实现执行指定次数的循环，运行结果如图 2.52 所示。



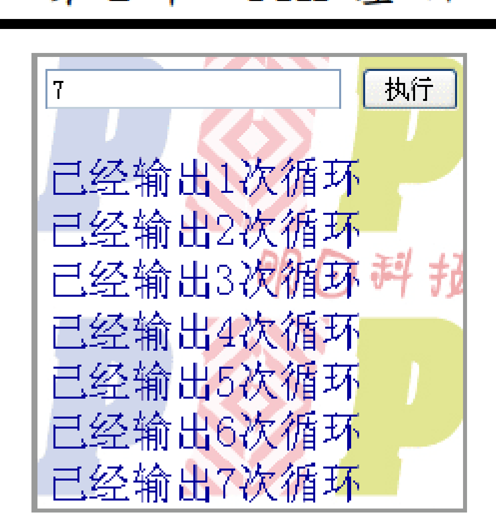


图 2.52　执行指定次数的循环

关键技术

本实例利用 break 语句可以跳出循环并终止循环的特点，控制循环执行的次数。

设计过程

（1）创建 PHP 脚本文件。当单击“执行”按钮时，首先，定义 for 循环语句，设定循环变量$a 的初始值为 0，并且每执行一次循环，$a 做一次自增运算。然后，通过 if 语句定义条件，如果$a 大于等于文本框中输入的数字时，跳出当前循环，否则输出字符串，其代码如下：

```
<?php
	if($_POST[sub1]){								//通过 POST 方式传递参数
		for($a = 0;$a < 10000000000000;$a++){		//循环
			if($a >= $_POST[text]){					//接收文本框信息判断条件
				break;								//结束循环
			}else{									//否则循环输出
				echo "已经输出".($a+1)."次循环<br>";
			}
		}
	}
?>
```

（2）将该文件存储于\MR\02\078 文件夹下，命名为 index.php。运行结果如图 2.52 所示。

秘笈心法

心法领悟 078：流程控制语句。

流程控制语句包括条件控制语句、循环控制语句和跳转控制语句，是 PHP 的重要组成部分，也是 PHP 程序员必须重点掌握和理解的内容。

2.7　自定义函数

在开发过程中，经常需要多次重复某种操作或处理，如数据查询、字符操作等。如果每个模块都要重新输入一次代码，不仅加大了程序员的工作量和开发时间，而且对于代码的后期维护及运算效果也有很大影响。使用自定义函数就是解决该问题的最有效的方法。

实例 079	自定义函数截取中文字符串 光盘位置：光盘\MR\02\079	高级 趣味指数：★★★★

实例说明

截取字符串在程序设计中是经常出现的，例如，截取增、删、改、查 SQL 语句的前 6 个字符进行判断，以便

进行增、删、改、查等操作。本实例应用 substr()函数实现用自定义函数截取中文字符串，运行结果如图 2.53 所示。

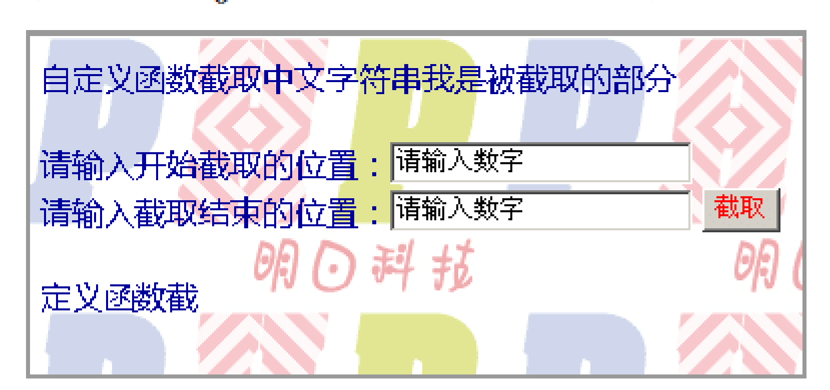


图 2.53　自定义函数截取中文字符串

关键技术

substr()函数用于从字符串中按照指定位置截取一定长度的字符，通过该函数可以获取某个固定格式字符串中的一部分。如果使用一个正数作为子串的起点来调用这个函数，将得到从起点到字符串结束的整个字符串；如果使用一个负数作为子串的起点来调用，将得到原字符串尾部的一个子串，字符个数等于给定负数的绝对值。substr()函数的语法如下：

```
string substr ( string str, int start [, int length]);
```

参数 str 用来指定字符串对象；参数 start 用来指定开始截取字符串的位置，如果为负数，则从字符串的末尾开始截取；参数 length 为可选项，指定截取字符的个数，如果为负数，则表示截取到倒数第 length 个字符。

设计过程

（1）创建 PHP 脚本文件，定义自定义函数，当单击“截取”按钮时，利用 substr()函数对指定字符串进行截取并输出截取信息，其代码如下：

```
<?php
        function str(){                                                        //自定义函数 str()
                if($_POST[sub]){                                               //通过 POST 方式传递参数
                        $a = "自定义函数截取中文字符串我是被截取的部分";          //定义字符串变量
                        $rs = substr($a,(2*$_POST['text']-2),(2*$_POST['text_1']));  //截取字符串
                        echo $rs;                                              //输出截取结果
                }
        }
str();                                                                         //自定义函数调用
?>
```

（2）将该文件存储于\MR\02\079 文件夹中，命名为 index.php。运行结果如图 2.53 所示。

秘笈心法

心法领悟 079：substr()函数的特点。

substr()函数支持从右侧取值，例如：

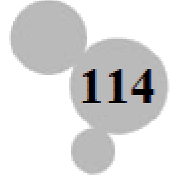
```
<?php
        $a ="abcdef";                 //定义字符串
        $rs = substr($a,-2,1);        //从右侧取值
        echo $rs;                     //输出截取结果
?>
```

经 IE 浏览器运行，显示结果为 e。

实例 080	公告标题的截取 光盘位置：光盘\MR\02\080	高级 趣味指数：★★★★

实例说明

本实例通过字符串截取函数和自定义函数实现公告标题的截取，运行结果如图 2.54 所示。



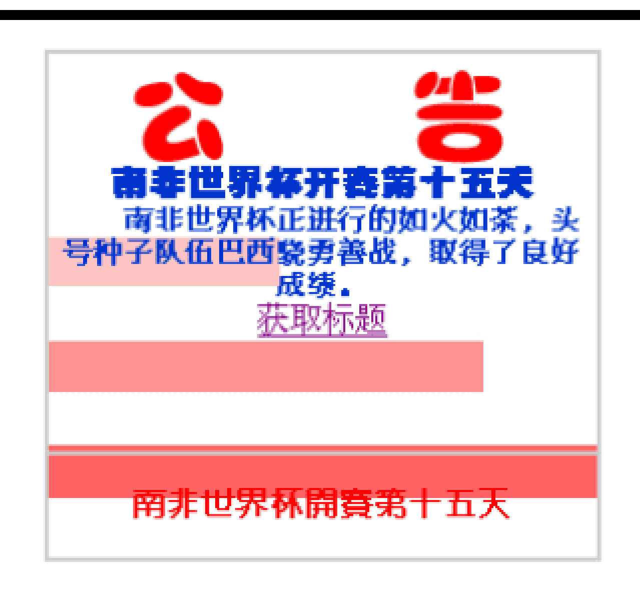


图 2.54　公告标题的截取

关键技术

自定义函数，顾名思义是由用户自行定义并完成指定功能的函数。自定义函数的产生是由于程序员在进行日常开发时，使用 PHP 预定义的函数并不能实现理想的功能。程序员迫切希望能够根据自己程序的实际情况自行定义完成指定功能的函数。创建函数的基本语法格式如下：

```
function function_name ([$arg_1],[$arg_2], ... ,[$arg_n]){
    fun_body;
[return arg_n;]
}
```

参数说明：

function：声明自定义函数时必须使用的关键字。

function _name：创建函数的名称，是有效的 PHP 标识符。函数名称是唯一的，其命名遵守与变量命名相同的规则，只是不能以“$”开头。

arg_1...arg_n：是外界传递给函数的值，可有可无，也可以有多个，数量根据需要而定，各参数用逗号“,”分隔。参数的类型不必指定，在调用函数时只要是 PHP 支持的类型都可以使用。

fun_body：自定义函数的主体，是功能实现部分。

return：将调用的代码需要的值返回，并结束函数的运行。

在本实例中，定义自定义函数 str()，对字符串进行截取操作，其语法如下：

```
function str($str){                          //自定义函数
        $a = strlen($str);                   //计算字符串长度
        $b = substr($str,0,$a);              //截取字符串
        echo $b;                             //输出
}
```

设计过程

（1）创建 PHP 脚本文件，定义数组，当单击“获取标题”超链接时，调用自定义函数 str()对数组中的元素进行截取，并输出截取后的标题信息，其代码如下：

```
<?php
 //定义数组
  $array = array("南非世界杯开赛第十五天","南非世界杯正进行得如火如荼，头号种子队伍巴西骁勇善战，取得了良好成绩。");
?>
<table border="1"  bordercolor="#CCCCCC"cellspacing="0" width="290" height="251" align="center"  background="2.jpg">
  <tr><td width="208" align="center">
  <b style="font-family:'文鼎霹雳体'; font-size:18px; color:#0033CC"><?php echo $array[0];?></b><br>
  <b style="font-family:'方正新报宋简体'; font-size:14px; color:#0033CC">      <?php echo
$array[1];?></b><br>
  <a href="index.php?sub=获取" style="float:center">获取标题</a>            //定义超链接
  </td>
</tr><tr><td align="center"><b style="color:#FF0000; font-family:'方正细圆繁体'">
<?php
     if($_GET[sub] == "获取"){                                    //通过 GET 方式获取参数
```

```
            str($array[0]);                                         //调用自定义函数
        }
        function str($str){                                         //自定义函数
            $a = strlen($str);                                      //计算字符串长度
            $b = substr($str,0,$a);                                 //截取字符串
            echo $b;                                                //输出
        }
?>
```

（2）将该文件存储于\MR\02\080 文件夹下，命名为 index.php。运行结果如图 2.54 所示。

秘笈心法

心法领悟 080：自定义函数的应用注意事项。

自定义函数的名称是不区分大小写的，而常量和变量的名称区分大小写。

实例 081 论坛内容的简短输出

光盘位置：光盘\MR\02\081

高级

趣味指数：★★★★

实例说明

开发论坛程序很重要的一点是转换空格符和回车符，因为不是所有的论坛用户都知道 HTML 标签“ ”和
的作用，大多数用户都喜欢用空格和回车对文本进行控制，这就需要程序员人为地做到统一。本实例通过自定义函数定义转换方法实现论坛内容的简短输出，运行结果如图 2.55 所示。

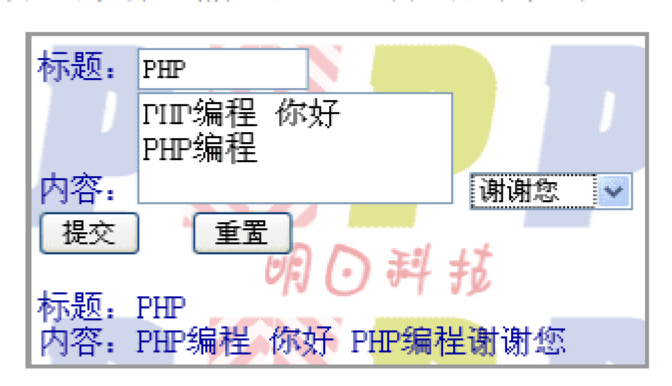


图 2.55　论坛内容的简短输出

关键技术

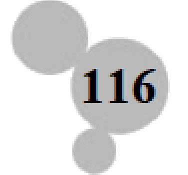
本实例通过自定义函数实现对字符串中空格符和回车符的转换输出，在自定义函数中应用的是 preg_replace()函数。自定义函数的语法如下：

```
function str($str){                                                 //自定义函数
        $str = preg_replace("/ /"," ",$str);                   //替换空格
        $str = preg_replace("/cha(13)/","<br>",$str);               //替换回车符
        echo "内容：".$str;                                          //输出
}
```

设计过程

（1）创建 PHP 脚本文件，通过自定义函数定义对“ ”和回车进行替换，当单击“提交”按钮时，首先对文本框是否存在内容进行判断，如果文本框内容不为空，调用自定义函数并输出文本信息，其代码如下：

```
<?php
        function str($str){                                         //自定义函数
            $str = preg_replace("/ /"," ",$str);               //替换空格
            $str = preg_replace("/cha(13)/","<br>",$str);           //替换回车符
            echo "内容：".$str;                                      //输出
        }
        if($_POST[sub]){                                            //通过 POST 方式传递参数
        //如果 text 为空或者文本域和下拉列表框同时为空
```



```
        if($_POST[text] == "" || (($_POST[te] == "") && ($_POST[check] == 1))){
            echo "<script>alert('内容不能为空');location.href='index.php'</script>";          //输出提示
        }else{
                if($_POST[select] != '1'){                                                  //当下拉列表框值不等于 1 时
                    echo "标题: ".$_POST[text]."<br>";                                       //输出内容和下拉列表框内容
                    str($_POST[te].$_POST[check]);
                }else{
                    echo "标题: ".$_POST[text]."<br>";                                       //否则只输出文本域内容
                    str($_POST[te]);
            }
        }
    }
?>
```

（2）将该文件存储于\MR\02\081 文件夹下，命名为 index.php。运行结果如图 2.55 所示。

秘笈心法

心法领悟 081：对于实例代码的重要说明。

上例代码中“$_POST[text] == "" || (($_POST[te] == "") && ($_POST[check] == 1))”是存在运算符优先级的，为了避免发生错误，笔者将先进行的运算加上了小括号。

实例 082　自定义函数过滤字符串

光盘位置：光盘\MR\02\082

高级

趣味指数：★★★★

实例说明

用户在论坛上发表或者回复留言时可能会遇到这种情况，单击“发布”按钮时，系统提示使用了禁用词语，禁止发布。这是论坛的开发人员为了规范论坛而采用的过滤字符串的手段。本实例通过自定义函数实现过滤字符串，运行结果如图 2.56 所示。

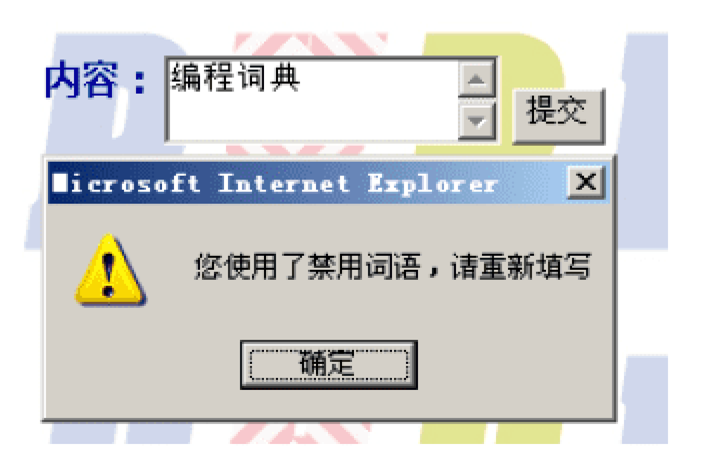


图 2.56　自定义函数过滤字符串

关键技术

本实例编写自定义函数 str()对字符串进行过滤，如果出现指定的关键词，则给出提示信息并终止程序执行。自定义函数 str()的语法如下：

```
function str($str){
    $array = array('图书','明日科技','软件','编程词典','编程','词典');          //定义数组
    $repstr = implode($array);                                               //数组转换成字符串
    if(preg_match("/$str/",$repstr)){                                        //正则表达式验证字符串
        echo "<script>alert('您使用了禁用词语，请重新填写');location.href='index.php'</script>";
    }else{
        echo "内容为: ".$str;                                                 //输出数据
    }
}
```

设计过程

（1）创建 PHP 脚本文件。首先定义自定义函数，将禁用词语保存在数组中，并通过字符串函数拆分数组，利用正则表达式函数对禁用词语进行屏蔽。当单击“提交”按钮时，调用自定义函数对提交内容进行过滤，并输出提示信息，其代码如下：

```
<?php
function str($str){
	$array = array('图书','明日科技','软件','编程词典','编程','词典');			//定义数组
	$repstr = implode($array);							//数组转换成字符串
	if(preg_match("/$str/",$repstr)){						//正则表达式验证字符串
		echo "<script>alert('您使用了禁用词语，请重新填写');location.href='index.php'</script>";
	}else{
		echo "内容为：".$str;							//输出数据
	}
}
if($_POST[sub]){
	str($_POST[te]);								//调用自定义函数
}
?>
```

（2）将该文件存储于\MR\02\082 文件夹下，命名为 index.php。运行结果如图 2.56 所示。

秘笈心法

心法领悟 082：过滤程序。

本实例只是简单地介绍了字符串过滤的原理，真正的过滤程序由于数据量较大，需要将数据存储在数据库中。如果读者感兴趣，可以将其与数据库中的数据结合使用。

2.8 字 符 串

在许多 Web 编程中，字符串总是会被大量地生成和处理。正确地使用和处理字符串，对于 PHP 程序员来说越来越重要。

实例 083	过滤论坛帖子中的空白和特殊字符 光盘位置：光盘\MR\02\083	高级 趣味指数：★★★★

实例说明

由于各个用户所使用的输入法不同或者个人习惯的原因，有可能不经意间多输入了空格或者特殊字符，造成网站内容的输出不美观。本实例通过 ltrim()和 rtrim()函数实现去除字符串中的空白和特殊字符，运行结果如图 2.57 所示。

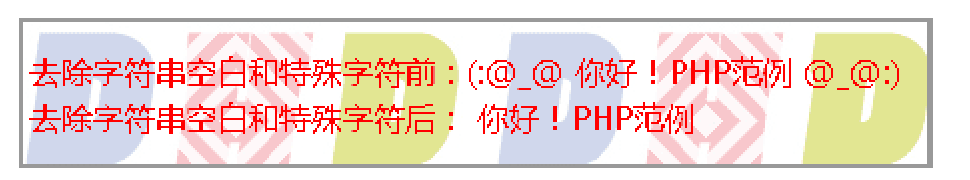
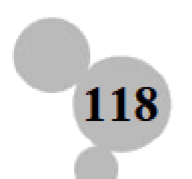


图 2.57　去除字符串中的空白和特殊字符

关键技术

（1）ltrim()函数

使用 ltrim()函数默认情况下去除左边空格，去除特殊字符的参数可选。



```
string ltrim ( string str [, string charlist]);
```

参数说明：

string str：源字符串信息。

string charlist：可选参数，特殊字符。

（2）rtrim()函数

使用 rtrim()函数默认情况下去除右边空格，去除特殊字符的参数可选。

```
string rtrim ( string str [, string charlist]);
```

参数说明：

string str：源字符串信息。

string charlist：可选参数，特殊字符。

设计过程

（1）创建 PHP 脚本文件，定义字符串变量$a，分别通过 ltrim()和 rtrim()函数去除两边的空格和特殊字符并显示结果，其代码如下：

```
<?php
	$a = "(:@_@   你好！PHP 范例   @_@:)";		//声明一个带空格和特殊字符的字符串变量
	$b = ltrim($a);					//去除变量左边的空格
	$c = ltrim($b,"(:@_@");				//去除变量左边的特殊字符
	$d = rtrim($c);					//去除变量右边的空格
	$e = rtrim($d,"@_@:)");				//去除变量右边的特殊字符
	echo "去除字符串空白和特殊字符前：".$a."<br>";
	echo "去除字符串空白和特殊字符后：".$e;		//输出最后结果
?>
```

（2）将该文件存储于\MR\02\083 文件夹下，命名为 index.php。运行结果如图 2.57 所示。

秘笈心法

心法领悟 083：使用 ltrim()和 rtrim()函数的相关说明。

在使用这两个函数时，默认情况下是只去除左边或者右边的空格，想要去除特殊字符，需要在函数里注明。

实例 084	对论坛帖子的内容进行转义 光盘位置：光盘\MR\02\084	高级 趣味指数：★★★★

实例说明

论坛帖子中的单引号和双引号在未经转义的情况下输出会出现问题，特别是在输出 SQL 语句时。本实例通过 addslashes()函数来实现转义字符串，运行结果如图 2.58 所示。

```
未经转义的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES('1','YangMing')
经转义后的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES(\'1\',\'YangMing\')
```

图 2.58　转义字符串

关键技术

本实例主要是通过 addslashes()函数对指定的 SQL 语句进行自动转义。addslashes()函数可以转义单引号、双引号、反斜杠、NULL 字符，主要用于 SQL 语句，对部分字符进行转义，其语法如下：

```
string addslashes ( string str);
```

参数说明：

string str：SQL 语句。

设计过程

（1）创建 PHP 脚本文件，首先定义插入型 SQL 语句，然后将 SQL 语句通过 addslashes()函数转义并显示转义结果，其代码如下：

```
<?php
    $str = "INSERT INTO db(id,user,pwd,last_date)VALUES('1','yang','******','2010-06-03')";     //声明一个字符串变量
    echo "未经转义的字符串：".$str."<br>";                                                    //未经转义
    echo "经转义后的字符串：".addslashes($str);                                                //经转义
?>
```

（2）将该文件存储于\MR\02\084 文件夹中，命名为 index.php。运行结果如图 2.58 所示。

秘笈心法

心法领悟 084：addslashes()函数在程序中的应用。

addslashes()函数主要用于数据库 SQL 语句的转义，当数据要插入至数据库之前，有必要应用 addslashes()函数进行字符串转义，以免特殊字符未经转义在插入数据库时出现错误。

实例 085 还原论坛中帖子的内容

光盘位置：光盘\MR\02\085

高级

趣味指数：★★★★

实例说明

既然可以将帖子内容进行转义，当然也可以将帖子内容进行还原。本实例通过 stripslashes()函数来实现还原字符串，运行结果如图 2.59 所示。

未经还原的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES(\'1\',\'YangMing\')
经还原后的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES('1','YangMing')

图 2.59　还原字符串

关键技术

本实例运用 stripslashes()函数，将 addslashes()函数转义的字符串进行还原，其语法如下：

```
string stripslashes ( string str);
```

参数说明：

string str：SQL 语句。

设计过程

（1）创建 PHP 脚本文件，首先定义数据库插入语句的字符串变量，然后将字符串变量通过 stripslashes()函数还原并显示运行结果，其代码如下：

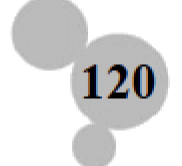
```
<?php
    $str = "INSERT INTO db(id,user,pwd,last_date)VALUES('1','yang','******','2010-06-03')";     //声明一个字符串变量
    echo "未经还原的字符串：".$str."<br>";                                                    //未经还原字符串输出
    echo "经还原后的字符串：".stripslashes($str)."<br>";                                      //经还原后的字符串输出
?>
```

（2）将该文件存储于\MR\02\085 文件夹下，命名为 index.php。运行结果如图 2.59 所示。

秘笈心法

心法领悟 085：stripslashes()函数的应用。

stripslashes()函数主要用于数据库 SQL 语句的还原输出。



实例 086　截取论坛标题　高级

光盘位置：光盘\MR\02\086　趣味指数：★★★★

实例说明

本实例通过 substr()函数实现截取论坛标题，运行结果如图 2.60 所示。

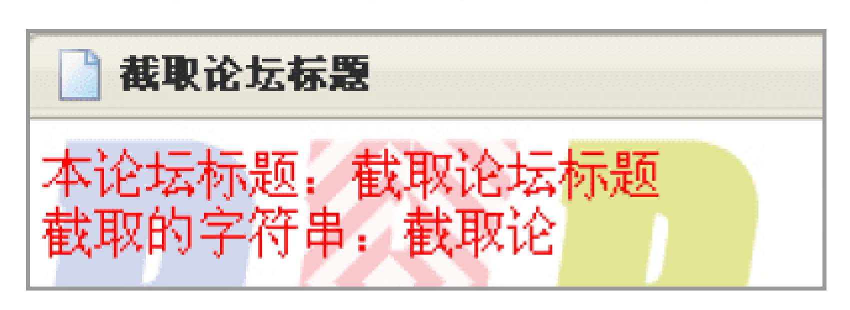


图 2.60　截取论坛标题

关键技术

substr()函数用于从字符串中按照指定位置截取一定长度的字符，通过该函数可以获取某个固定格式字符串中的一部分。substr()函数的详细讲解请参见实例 079。

设计过程

（1）创建 PHP 脚本文件。首先定义变量$a 并将网页标题赋给$a，然后通过 substr()函数对字符串$a 进行截取，并输出截取内容，其代码如下：

```
<?php
	$b = "截取论坛标题";                              //设定字符串
	$a = "<title>$b</title>";                         //设定网页标题
	echo $a;                                          //输出字符串
	echo "本论坛标题：".$b."<br>";                    //输出论坛标题
	$b = substr($b,0,12);                             //截取论坛标题
	echo "截取的字符串：".$b;                          //输出截取内容
?>
```

（2）将该文件存储于\MR\02\086 文件夹下，命名为 index.php。运行结果如图 2.60 所示。

秘笈心法

心法领悟 086：substr()函数应用的注意事项。

substr()函数的参数 start 的指定位置是从 0 开始计算的，即字符串中的第 1 个字符表示为 0。该函数在截取中文字符串时可能会出现乱码，所以建议在对中文字符串进行截取时应用 mb_substr()函数。

实例 087　分割、合成字符串　高级

光盘位置：光盘\MR\02\087　趣味指数：★★★★

实例说明

字符串的分割、合成对于程序员来说是十分重要的，应用也十分广泛。例如，在遍历数据库查找匹配项时就会用到合成函数，而购物车程序一般会用到分割函数。本实例通过 explode()函数和 implode()函数实现分割、合成字符串，运行结果如图 2.61 所示。

原字符串：你好PHP 很高兴认识你。
经分割后的数组为：Array ([0] => 你好PHP [1] => 很高兴认识你。)
合成后的字符串：你好PHP很高兴认识你。

图 2.61　分割、合成字符串

关键技术

字符串的分割与合成应用的是 explode()函数和 implode()函数。

（1）explode()函数

explode()函数按照指定的规则对一个字符串进行分割，返回值为数组，其语法如下：

```
array explode ( string separator, string string [, int limit]);
```

参数 separator 表示指定的分隔符；string string 表示分割条件；int limit 表示指定生成数组的最大值。

（2）implode()函数

implode()函数与 explode()函数配合使用，将数组的内容组合成一个字符串，其语法如下：

```
string implode ( string glue, array pieces);
```

参数 glue 是字符串类型，指定分隔符；参数 pieces 是数组类型，指定要被合并的数组。

设计过程

（1）创建 PHP 脚本文件。首先定义字符串变量，将字符串内容用指定的分隔符分隔。然后利用 explode()函数，将字符串变量按照规定的分隔符分割，并保存在数组变量$explode 中。最后分别利用 prinr_r()函数和 implode()函数输出数组和字符串，其代码如下：

```
<?php
        $str = "你好 PHP  很高兴认识你。";                    //定义字符串
        $explode = explode(" ",$str);                         //分割字符串
        echo "原字符串：".$str."<br>";                         //输出原字符串
        echo "经分割后的数组为：";
        print_r($explode);                                    //打印分割后的数组
        echo "<br>合成后的字符串：";
        echo implode($explode);                               //将数组合成为字符串
?>
```

（2）将该文件存储于\MR\02\087 文件夹下，命名为 index.php。运行结果如图 2.61 所示。

秘笈心法

心法领悟 087：explode()函数和 implode()函数哪个应用更广泛。

二者相比，implode()函数应用相对广泛，所有的数组变量都可以用它进行合成。

实例 088	查询关键字描红 光盘位置：光盘\MR\02\088	高级 趣味指数：★★★★

实例说明

关键字描红技术广泛应用于站内搜索或者高级搜索中，是一项很常用的技术。本实例通过 str_replace()函数实现查询关键字描红，运行结果如图 2.62 所示。

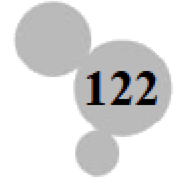


图 2.62　查询关键字描红

关键技术

str_replace ()函数可用于执行字符串替换操作，其语法如下：

```
mixed str_replace ( mixed search, mixed replace, mixed subject [, int &count] )
```

参数说明：

search：必要参数，指定需要查找的字符串。

replace：必要参数，指定替换的值。

subject：必要参数，指定查找范围。

count：可选参数，获取执行替换的数量。

设计过程

（1）创建 PHP 脚本文件，当单击“搜索”按钮时，将通过 POST 方式接收到数据，首先对数据进行描红处理，然后应用 str_replace()函数对指定字符串信息的指定内容进行替换，其代码如下：

```
<?php
if($_POST[sub]){
        $a = "<b style='color:red;font-size:18px;'>".$_POST[text]."</b>";          //对提交的关键字进行描红
        echo str_replace("$_POST[text]",$a,$str);                               //完成字符串的替换操作
}
?>
```

（2）将该文件存储于\MR\02\088 文件夹下，命名为 index.php。运行结果如图 2.62 所示。

秘笈心法

心法领悟 088：更多实现关键字描红的函数。

不仅可以通过 str_replace()函数实现关键字的描红操作，还可以通过 str_ireplace()函数实现对关键字的描红功能，并且该函数不区分大小写。

实例 089　统计查询关键字出现的次数　高级

光盘位置：光盘\MR\02\089　趣味指数：★★★★

实例说明

在站内搜索中，往往要求列出符合条件的关键字有多少个。本实例通过字符串函数 substr_count()统计查询关键字出现的次数，运行结果如图 2.63 所示。

图 2.63　统计查询关键字出现的次数

关键技术

本实例通过 substr_count()函数检索字符串子串出现的次数。substr_count()函数的语法如下：

```
int substr_count ( string haystack, string needle )
```

参数说明：

string haystack：源字符串。

string needle：字符串子串。

设计过程

（1）创建 PHP 脚本文件，首先定义字符串变量，编写 form 表单。当单击“搜索”按钮时，利用 substr_count()函数输出指定字符串子串的出现次数，其代码如下：

```
<?php
        echo "<a style='font-size:20px; color:#000000'>被搜索的文本：</a>";          //定义字符串输出
        $str = "PHP 作为全球最普及、应用最广泛的互联网开发语言之一，从 1994 年诞生至今已被 2000 多万个动态网站采用，全球知名
互联网公司 Google、Yahoo、eBay 和中国知名网站新浪、百度、阿里巴巴等均采用 PHP 技术！";
        echo $str."<br>";
?>
<form action ="" method="post">                                                   //表单
    关键字：<input type="text" name="text"><input type="submit" name="sub" value ="搜索">
</form>
</body>
</html>
<?php
    if($_POST[sub]){                                                             //通过 POST 方式传递参数
        $a = "<b style='color:red;font-size:18px;'>".$_POST[text]."</b>";         //替换后的字符串
        echo str_replace("$_POST[text]",$a,$str)."<br>";                          //替换字符串
        //统计关键字出现次数
        echo "关键字出现<b style='color:red;font-size:22px;'>".substr_count($str,$_POST[text])."</b>次。";
    }
?>
```

（2）将该文件存储于\MR\02\089 文件夹下，命名为 index.php。运行结果如图 2.63 所示。

秘笈心法

心法领悟 089：substr_count()函数的实现原理。

substr_count()函数是一种类似全文检索的函数，其原理是检索查找匹配的字符串，返回匹配次数。

实例 090	获取上传文件的后缀 光盘位置：光盘\MR\02\090	高级 趣味指数：★★★★

实例说明

文件上传功能几乎是所有网站所必备的，然而文件上传对于服务器来说具有很大风险，因此应该对文件大小、文件类型进行限制。本实例通过字符串函数 strrev()获取上传文件的后缀，运行结果如图 2.64 所示。

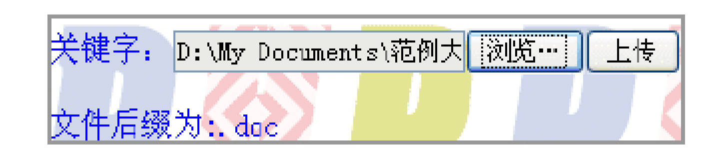


图 2.64　获取上传文件的后缀

关键技术

PHP 中字符串的逆序输出可以通过循环实现，但最简单的方式是通过 strrev()函数实现。strrev()函数可将字



符串反转，其语法如下：

```
string strrev ( string string );
```

参数 string 为需要反转的字符串。

在本实例中，通过 strrev()函数对上传文件的名称进行反转输出，并通过 explode()函数以“.”为分隔符，对文件名称进行分割，然后再次应用 strrev()函数对数组中的第一个元素值进行返回输出，获取到的就是上传文件的后缀。

设计过程

（1）创建 PHP 脚本文件，当单击“上传”按钮时，首先利用 POST 方法接收文本框中的信息，然后利用反转函数将文本框信息数据进行反转，并利用字符串拆分函数以“.”进行分割，将分割结果保存在数组中。最后将指定数据再次反转取得并输出上传文件的后缀名称，其代码如下：

```
<?php
    if($_POST[sub]){                              //通过 POST 方式传递参数
        $a = strrev($_POST[text]);                //反转字符串
        $b = explode(".",$a);                     //以点号分割
        $c = strrev($b[0]);                       //反转字符串
        echo "文件后缀为:.$c";                    //输出文件后缀
    }
?>
```

（2）将该文件存储于\MR\02\090 文件夹下，命名为 index.php。运行结果如图 2.64 所示。

秘笈心法

心法领悟 090：知识扩展。

利用 strrev()函数检索字符串相对较麻烦，可以使用正则表达式函数 preg_match()来完成该功能。

实例 091	统一上传文件名称的大小写 光盘位置：光盘\MR\02\091	高级 趣味指数：★★★★

实例说明

统一上传文件名称的大小写是十分必要的。例如上传图片文件 img.jpg 和 Img.jpg，如果文件名称未经大小写统一，则上传时可能不会出现覆盖提示。本实例通过字符串函数 strtoupper()和 strtolower()实现统一上传文件名称的大小写，运行结果如图 2.65 所示。

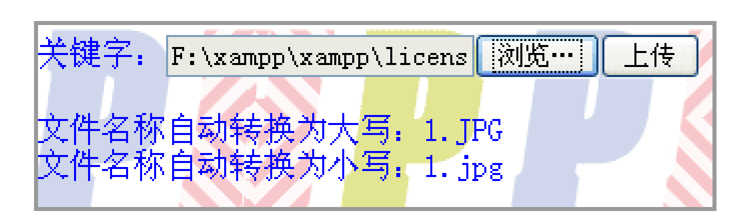


图 2.65　统一上传文件名称的大小写

关键技术

实现字符串的大小写转换，主要通过 strtolower()和 strtoupper()函数实现。

（1）strtolower()函数

使用该函数转换为小写。

```
string strtolower ( string str )
```

（2）strtoupper()函数

使用该函数转换为大写。

```
string strtoupper ( string string )
```

参数 string 为需要转换的字符串。

设计过程

（1）创建 PHP 脚本文件，当单击“上传”按钮时，分别利用转换小写和大写的函数 strtolower()和 strtoupper()将接收到的文本框信息进行对应的大小写转换，并输出转换结果，其代码如下：

```
<?php
	if($_POST[sub]){                                          //通过 POST 方式传递参数
		$a = strtoupper($_POST[text]);                         //转换为大写
		echo "文件名称自动转换为大写：".$a."<br>";             //输出
		$b = strtolower($_POST[text]);                         //转换为小写
		echo "文件名称自动转换为小写：".$b;                    //输出
	}
?>
```

（2）将该文件存储于\MR\02\091 文件夹下，命名为 index.php。运行结果如图 2.65 所示。

秘笈心法

心法领悟 091：strtolower()和 strtoupper()函数的应用范围。

在电子商务类网站的后台商品添加页面中，为了使商品的型号以统一的规格显示，经常会使用大小写转换函数，所以掌握这两个函数是很有必要的。

实例 092	论坛中直接输出 HTML 脚本 光盘位置：光盘\MR\02\092	高级 趣味指数：★★★★

实例说明

在论坛文本框里使用 HTML 标签是存在作用效果的，但是对于论坛管理员来说是不希望看到的，因为它会给服务器造成压力。本实例通过 htmlentities()函数实现字符串与 HTML 的转换，运行结果如图 2.66 所示。



图 2.66　字符串与 HTML 转换

关键技术

PHP 中使 HTML 代码格式化输出可以使用函数 htmlentities()实现。htmlentities()函数的语法如下：

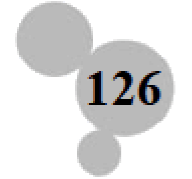
```
string htmlentities(string string,[int quote_style],[string charset]);
```

参数说明：

string string：需要格式化输出的 HTML 代码。

int quote_style：可选参数，一般设置为 ENT_QUOTES。

string charset：设置编码格式。



设计过程

（1）创建 PHP 脚本文件，首先，定义字符串变量，变量内容为一段 HTML 标签（创建表格的代码）。然后，利用 htmlentities()函数将字符串变量进行转换并输出转换结果，其代码如下：

```
<?php
	echo "<h2>没有进行转换</h2>";
$str = "<table border = '1'><tr><td>标题：经济半小时</td><td>修改时间：2010-06-03</td></tr><tr><td>内容：经济半小时，了解中国经济最好的节目</td></tr></table>";		//定义 HTML 格式
	echo $str;									//输出字符串
	echo "<h2>进行了转换</h2>";
	echo htmlentities($str,ENT_QUOTES,"GB2312")."<br>";		//转换 HTML
?>
```

（2）将该文件存储于\MR\02\092 文件夹下，命名为 index.php。运行结果如图 2.66 所示。

秘笈心法

心法领悟 092：使用格式化函数输出的优点。

格式化输出 HTML 可以直接将提交的源码输出并确保源码不被解析，从而可以确保直接在页面中输出 HTML 标签，而不被解析。

实例 093	统计帖子标题的长度 光盘位置：光盘\MR\02\093	高级 趣味指数：★★★★

实例说明

统计字符串长度一般是为了给其他函数的应用做好铺垫。本实例通过 strlen()函数获取帖子标题的长度，运行结果如图 2.67 所示。

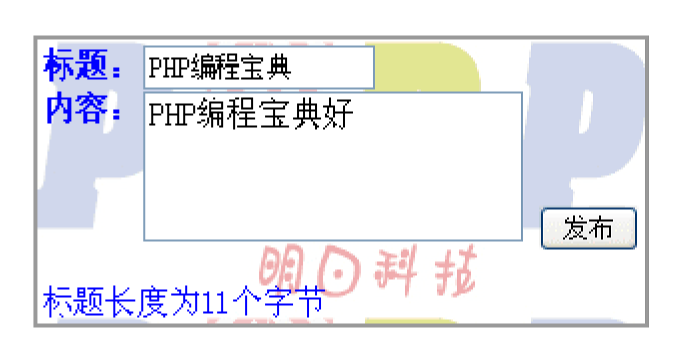


图 2.67　统计帖子标题的长度

关键技术

strlen()函数用于获取字符串的长度，其语法如下：

```
int strlen ( string str);
```

参数说明：

str：指定字符串。

设计过程

（1）创建 PHP 脚本文件，当单击“发布”按钮时，利用 strlen()函数获取通过 POST 方式接收到的数据长度，其代码如下：

```
<?php
	if($_POST[sub]){						//通过 POST 方式传递参数
		echo "标题长度为".strlen($_POST[text])."个字节";	//获取字符串长度
	}
?>
```

（2）将该文件存储于\MR\02\093 文件夹下，命名为 index.php。运行结果如图 2.67 所示。

秘笈心法

心法领悟 093：使用 strlen()函数的注意事项。

使用 strlen()函数获取字符串长度时，如果字符串中存在空格，则空格也会被计算在内。

实例 094	计算论坛帖子的字节数 光盘位置：光盘\MR\02\094	高级 趣味指数：★★★★

实例说明

使用 strlen()函数可以统计论坛标题的长度，同样可以对论坛中帖子的字节数进行统计。本实例通过 strlen()函数对论坛中帖子的字节数进行统计，运行结果如图 2.68 所示。

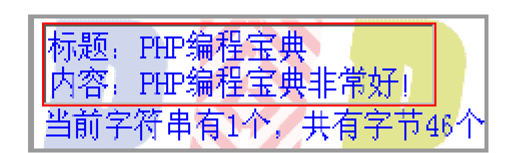


图 2.68　计算论坛帖子的字节数

关键技术

本实例同样应用 strlen()函数统计字符串的长度，并且应用自定义函数对提交的源代码进行处理，保证源代码以原形式输出，而不会输出执行文件。自定义函数的语法如下：

```
function unhtml($content){
     $content=htmlspecialchars($content);
     $content=str_replace(chr(13),"<br>",$content);
     $content=str_replace(chr(32)," ",$content);
    return trim($content);
}
```

设计过程

（1）创建 PHP 脚本文件，首先定义自定义函数，然后定义字符串变量，接着应用 strlen()函数统计帖子内容的字节数，并且通过自定义函数对帖子内容进行转换，其代码如下：

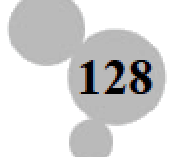
```
<?php
function unhtml($content){
     $content=htmlspecialchars($content);
     $content=str_replace(chr(13),"<br>",$content);
     $content=str_replace(chr(32)," ",$content);
    return trim($content);
 }
$str = "标题：PHP 编程宝典<br>内容：PHP 编程宝典非常好！";
?>
<table border="1" bordercolor="#FF0000" cellspacing="0"><tr><td>
<?php echo $str;?>
</td></tr></table>
<?php
echo "共有字节".strlen(unhtml($str))."个";
?>
```

（2）将该文件存储于\MR\02\094 文件夹下，命名为 index.php。运行结果如图 2.68 所示。

秘笈心法

心法领悟 094：自定义函数 unhtml()。

自定义函数 unhtml()主要用于对提交的源代码进行处理，保证源代码以原形式输出，而不会输出执行文件。



实例 095　实现帖子内容的定位查找

光盘位置：光盘\MR\02\095

高级

趣味指数：★★★★

实例说明

用微软 Windows 操作系统查找文件，可以使用 Ctrl+F 快捷键实现快速定位查找。本实例通过 strpos()函数模拟这一功能实现定位查找，运行结果如图 2.69 所示。

吉林省明日科技有限公司是一家以计算机软件技术为核心的高科技型企业

明日科技　定位查找

吉林省明日科技有限公司是一家以计算机软件技术为核心的高科技型企业
查找内容从6字节开始，字符串长度为8个字节

图 2.69　实现帖子内容定位查找

关键技术

在 PHP 中实现特殊字符的查找和定位可以通过 strpos()函数来实现，函数语法如下：

```
int strops(string str1 ,string str2, [int offset])
```

该函数的作用是在 str1 中查找 str2，并返回查找到的位置。strpos()函数实现的原理是比较整个要查找的字符串，并返回字符串 str2 第 1 次出现的位置。

设计过程

（1）创建 PHP 脚本文件，当单击“定位查找”按钮时，首先定义一个字符串变量$email，然后将接收到的文本框数据信息进行描红处理，最后利用 strpos()函数实现帖子内容的定位查找，其代码如下：

```
<?php
	if($_POST[sub]){
		$email = '吉林省明日科技有限公司是一家以计算机软件技术为核心的高科技型企业';
		$str = $_POST[te];
		$str1 = "<b style='color:red;'>".$str."</b>";
		echo str_replace($str,$str1,$email)."<br>";
		echo "查找内容从<b style='color:red;'>".strpos($email,$str)."</b>字节开始，字符串长度为<b style='color:red;'>".strlen($str)."</b>个字节";
	}
?>
```

（2）将该文件存储于\MR\02\095 文件夹下，命名为 index.php。运行结果如图 2.69 所示。

秘笈心法

心法领悟 095：知识扩展。

strpos()函数一般与 strlen()函数联合使用，得到的两个结果相当于 substr()函数的两个位置参数。

实例 096　去除帖子标题的首尾空格

光盘位置：光盘\MR\02\096

高级

趣味指数：★★★★

实例说明

用户在编写代码时也许会在不经意间多输入了一个或多个空格，使得程序无论怎么调试都无法正常运行。本实例通过 trim()函数实现去除帖子标题的首尾空格，运行结果如图 2.70 所示。

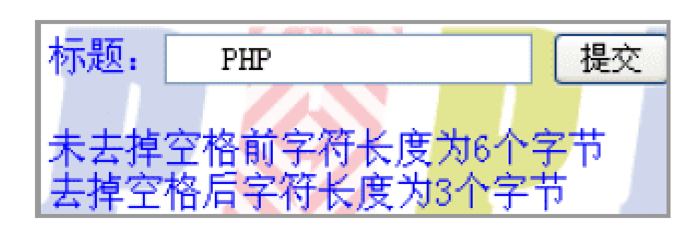


图 2.70　去除帖子标题的首尾空格

关键技术

在 PHP 中去除字符串的首尾空格，可以使用函数 trim()实现。trim()函数的语法如下：

```
string trim(string str);
```

参数说明：

string str：需要过滤空格的字符串。

设计过程

（1）创建 PHP 脚本文件，当单击“提交”按钮时，利用 strlen()和 trim()函数的联合操作，实现输出去除首尾空格前、后的字符串的长度，其代码如下：

```
<?php
	if($_POST[sub]){                                                    //通过 POST 方式传递参数
		echo "未去掉空格前字符长度为".strlen($_POST[text])."个字节<br>";      //输出结果
		echo "去掉空格后字符长度为".strlen(trim($_POST[text]))."个字节";       //输出结果
	}
?>
```

（2）将该文件存储于\MR\02\096 文件夹下，命名为 index.php。运行结果如图 2.70 所示。

秘笈心法

心法领悟 096：知识扩展。

PHP 还提供有单纯去除字符串左边或者右边空格的函数，详细说明参见实例 083。

实例 097	货币数据的格式化输出 光盘位置：光盘\MR\02\097	高级 趣味指数：★★★★

实例说明

货币数据不同于整型数据，其是存在一定格式的。本实例通过 number_format()函数实现金额的格式化输出，运行结果如图 2.71 所示。

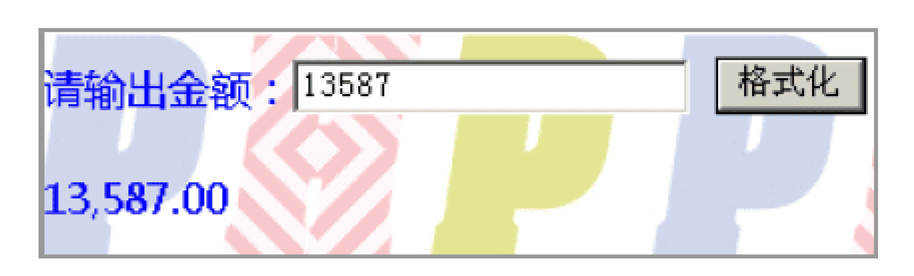


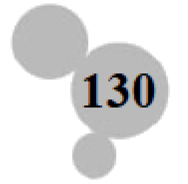
图 2.71　金额的格式化输出

关键技术

金额的格式化输出通过 number_format()函数实现，其语法及说明如下：

```
string number_format ( float number , int decimals)
string number_format ( float number, int decimals, string dec_point, string thousands_sep)
```

number_format()函数返回参数 number 格式化后的字符串，该函数可以有 1 个、2 个或 4 个参数，但不能是 3 个参数。如果只有 1 个参数 number，number 格式化后会舍去小数点后的值，且以千分位逗号“,”隔开；如



果有 2 个参数，number 格式化后会到小数点第 decimals 位，且会以每一千逗号来隔开；如果有 4 个参数，number 格式化后会到小数点第 decimals 位，dec_point 用来替代小数点“.”，thousands_sep 用来替代每一千隔开的逗号“,”。

设计过程

（1）创建 PHP 脚本，当单击“格式化”按钮时，程序自动将文本框中的信息进行格式化，代码如下：

```
<?php
	if($_POST[sub]){                                    //通过 POST 方式传递参数
		echo number_format($_POST[text],2);         //格式化数字
	}
?>
```

（2）将该文件存储于\MR\02\097 文件夹下，命名为 index.php。运行结果如图 2.71 所示。

秘笈心法

心法领悟 097：函数应用。

开发电子商务类网站， number_format()函数出现的频率是百分之百的，所以掌握此函数是十分必要的。

实例 098　日期、时间的格式化输出

光盘位置：光盘\MR\02\098

高级　趣味指数：★★★★

实例说明

在 PHP 程序设计中，经常用到对日期、时间的格式化输出。本实例通过 date()函数实现日期、时间的格式化输出，运行结果如图 2.72 所示。

显示当前日期、时间 2010-06-24 13:31:04

图 2.72　日期、时间的格式化输出

关键技术

本实例的关键点是 date()函数的灵活运用，详细说明详见实例 035。

设计过程

（1）创建 PHP 脚本，当单击“显示当前日期、时间”按钮时，利用 date()函数根据传入的指定的参数格式实现格式化输出，代码如下：

```
<?php
	date_default_timezone_set("Asia/ShangHai");        //设置时区
	If($_GET[date] == 1){                               //通过 GET 方式取得参数
		echo date("Y-m-d H:i:s");                   //格式化输出时间和日期
	}
?>
```

（2）将该文件存储于\MR\02\098 文件夹下，命名为 index.php。运行结果如图 2.72 所示。

秘笈心法

心法领悟 098：获取当地时间。

获取当地时间时一定要首先设置时区，否则 PHP 默认使用的是英国伦敦的零时区，而在本地使用的是北京时间，即东八区的时间。设置时区时，可使用 date_default_timezone_set("Asia/ShangHai")语句。

实例 099 比对论坛中帖子的相似度

光盘位置：光盘\MR\02\099

高级

趣味指数：★★★★

实例说明

在论坛程序中比较两个帖子是否相同可以屏蔽重复帖子，节省资源空间。本实例通过字符串函数 similar_text()比较两个帖子的相似度，运行结果如图 2.73 所示。

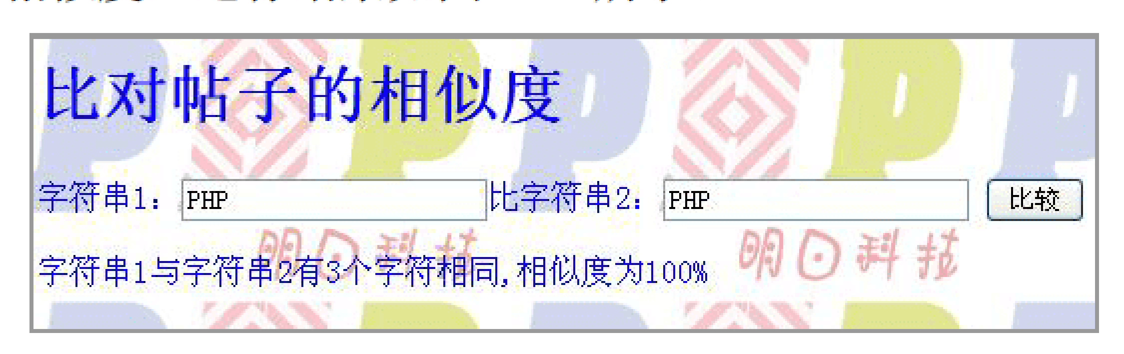


图 2.73　比对论坛中帖子的相似度

关键技术

PHP 中字符串相似度的获取可以通过 similar_text()函数来实现。其语法格式如下：

```
int similar_text(sting str1,string str2,[double precent])
```

该函数用于比较字符串 str1 和字符串 str2 的相似程度，函数的返回结果是字符串 str1 和字符串 str2 的相同字符的个数，而可省参数 precent 的值是这两个参数的相似度。

设计过程

（1）创建 PHP 脚本，当单击“比较”按钮时，首先，分别获取字符串 1 和字符串 2 中的数据，然后，利用字符串函数 simlar_text()实现字符串相似度的获取，代码如下：

```
<?php
	if($_POST[sub]){                                              //通过 POST 方式接收参数
		$a = similar_text($_POST[te],$_POST[tex]);                //判断两个字符串的相似度
		if(strlen($_POST[te]) >= strlen($_POST[tex])){            //计算百分比
			echo "字符串 1 与字符串 2 有".$a."个字符相同,相似度为".($a/strlen($_POST[te])*100)."%";
		}else{
			echo "字符串 1 与字符串 2 有".$a."个字符相同,相似度为".($a/strlen($_POST[tex])*100)."%";
		}
	}
?>
```

（2）将该文件存储于\MR\02\099 文件夹下，命名为 index.php。运行结果如图 2.73 所示。

秘笈心法

心法领悟 099：similar_text()函数的原理。

similar_text()函数的原理是对应比较每个字节并根据结果计算百分比。

实例 100 对用户注册的密码进行加密和解密

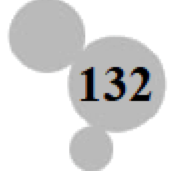
光盘位置：光盘\MR\02\100

高级

趣味指数：★★★★

实例说明

对于电子商务网站的用户来说，最重要的就是账号密码的安全性。如果账号密码未经过加密处理，则会大



大增加安全隐患，给非法用户以可乘之机。本实例是通过字符串函数 base64_encode()和 base64_decode()实现对密码的加密和解密，运行结果如图 2.74 所示。



图 2.74　对用户注册的密码进行加密和解密

关键技术

对字符串的 base64 编码可以通过 PHP 的预定义函数 base64_encode()实现，其语法如下：

```
string base64_encode(string str)
```

该函数实现对字符串 str 的 base64 编码。

对已经进行 base64 编码的字符串进行解码用函数 base64_decode()实现，其语法如下：

```
string base64_decode(string str)
```

该函数实现对已经进行 base64 编码的字符串进行解码。

设计过程

（1）创建 PHP 脚本，当单击“提交”按钮时，通过字符串函数 base64_encode()和 base64_decode()实现对字符串的加密和解密，代码如下：

```
<?php
	if($_POST[sub]){                              //通过 POST 方式接收参数
		echo "经加密<hr>";
		echo "用户名：".$_POST[text];
		$str = base64_encode($_POST[pwd]);        //对密码进行加密
		echo "密码：". $str;
		echo "<hr>";
		echo "经解密";
		echo "<hr>";
		echo "用户名：".$_POST[text];
		echo "密码：". base64_decode($str);        //对密码进行解密
	}
?>
```

（2）将该文件存储于\MR\02\100 文件夹下，命名为 index.php。运行结果如图 2.74 所示。

秘笈心法

心法领悟 100：加密函数的应用。

base64_encode()加密技术在邮件系统建设等方面也得到了广泛的应用。

实例 101	保护 URL 地址中传递的参数 光盘位置：光盘\MR\02\101	高级 趣味指数：★★★★

实例说明

表单的提交方式有两种：GET 方式和 POST 方式。GET 方式属于明文地址栏传参，存在安全隐患。本实例

通过 urlencode()函数实现保护 URL 地址中传递的参数，运行结果如图 2.75 所示。

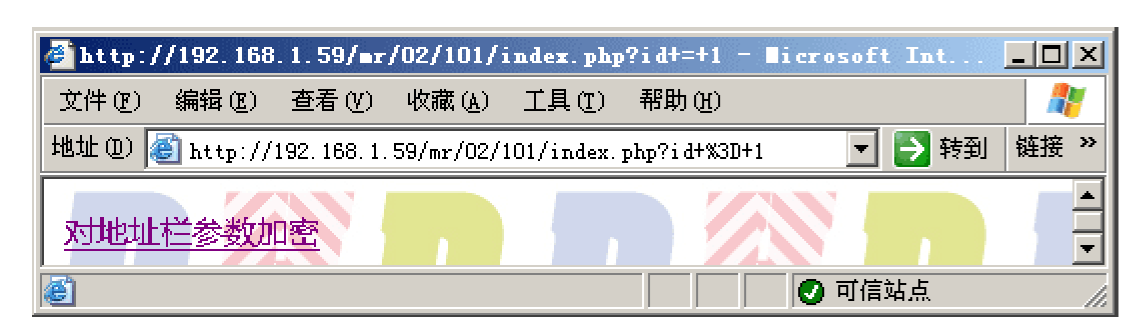


图 2.75　保护 URL 地址中传递的参数

关键技术

PHP 中字符串的 URL 编码通过 urlencode()函数实现，其语法如下：

```
string urlencode(string str)
```

该函数实现对字符串 str 的 URL 编码。

设计过程

（1）创建 PHP 脚本，定义字符串变量，将此字符串变量通过 urlencode()函数进行编码，然后将编码后的字符串变量以拼接地址栏参数的形式定义在超链接中，代码如下：

```
<?php
	$str = "id = 1";                                    //定义字符串
	$str = urlencode($str);                             //对字符串进行编码
?>
<a href="index.php?<?php echo $str;?>">对地址栏参数加密</a>        //链接
```

（2）将该文件存储于\MR\02\101 文件夹下，命名为 index.php。运行结果如图 2.75 所示。

秘笈心法

心法领悟 101：知识拓展。

经编码的 URL 参数是可以通过 urldecode()函数进行解码的，详细说明详见本实例。即使不对 URL 编码后的参数进行解码，同样也可以获取到传递的参数值。

实例 102　解析 URL 地址中传递的编码参数

光盘位置：光盘\MR\02\102

高级

趣味指数：★★★★

实例说明

本实例通过 urldecode()函数解析 URL 地址中传递的参数值，运行结果如图 2.76 所示。

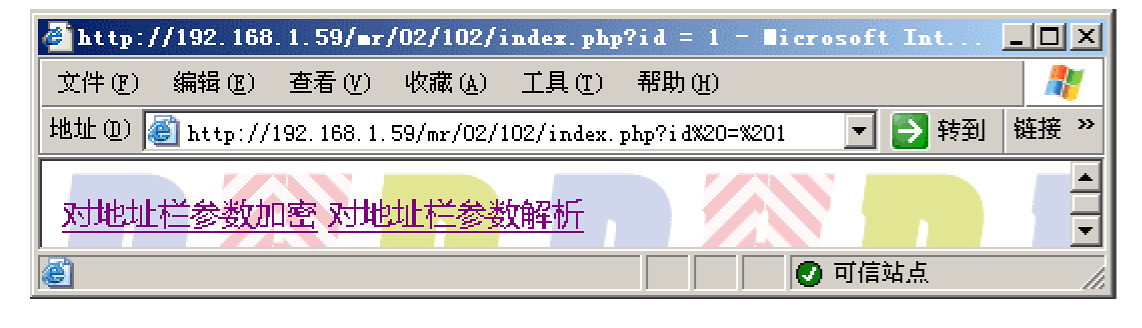
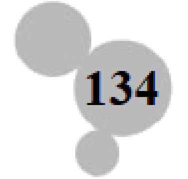


图 2.76　解析 URL 地址中传递的编码参数

关键技术

对已经进行 URL 编码的字符串进行解码使用 urldecode()函数，语法如下：

```
string urldecode ( string str);
```



设计过程

（1）创建 PHP 脚本。首先，定义字符串变量并为此字符串变量赋值。然后，将字符串变量进行 URL 编码处理。最后，单击超链接，并将链接地址用 urldecode()函数解码，代码如下：

```
<?php
	$str = "id = 1";                                          //定义字符串
	$str = urlencode($str);                                   //对 URL 进行编码
	$str1 = urldecode($str);                                  //对 URL 进行解析
?>
<a href="index.php?<?php echo $str1;?>">对地址栏参数解析</a>     //地址栏传参
```

（2）将该文件存储于\MR\02\102 文件夹下，命名为 index.php。运行结果如图 2.76 所示。

秘笈心法

心法领悟 102：URL 编码是否可以应用于防止 SQL 注入。

URL 编码不能有效地防止 SQL 注入，原因在于 URL 编码可以解析。

实例 103　转义帖子中的特殊字符

光盘位置：光盘\MR\02\103

高级　趣味指数：★★★★

实例说明

实例 043 已经向用户介绍过“\”转义特殊字符的方法。下面通过 quotemeta()函数实现转义特殊字符，运行结果如图 2.77 所示。

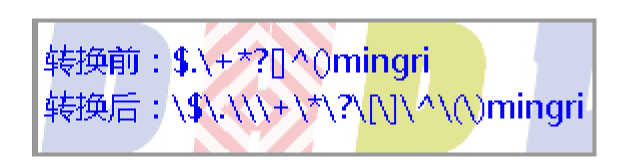


图 2.77　特殊字符的转换

关键技术

为了方便特殊字符的原样输出，PHP 提供了 quotemeta()函数，其语法如下：

```
string quotemeta(string str)
```

该函数的作用是在字符串 str 中的某些字符前面加上反斜杠“\”，这些特殊字符主要有“.”、“\”、“+”、“*”、“?”、“[”、“]”、“^”、“(”、“)”、“$”，其中，如果“$”后面的字符与“$”共同构成一个合法的变量名，则“$”前不加“\”。

设计过程

（1）创建 PHP 脚本。首先，定义字符串变量并将一系列特殊符号赋给变量。然后，通过 quotemeta()函数将字符串变量进行转义输出，代码如下：

```
<?php
	$s="mingri";                                              //定义字符串变量
	$str="$.\+*?[]^()$s";                                     //定义字符串变量
		echo"转换前：".$str."<br>";
		echo "转换后：".quotemeta($str);                      //转换特殊字符
?>
```

（2）将该文件存储于\MR\02\103 文件夹下，命名为 index.php。运行结果如图 2.77 所示。

秘笈心法

心法领悟 103：解决无法原样输出字符的问题。

在进行 PHP 项目开发过程中，经常会遇到一些无法原样输出的字符，解决办法就是通过转义字符进行转义。

实例 104 获取任意字符的 ASCII 码

光盘位置：光盘\MR\02\104

高级

趣味指数：★★★★

实例说明

敲击计算机键盘上的任意一个按键都要经过 ASCII 码转换后显示到屏幕上。本实例通过字符串函数 ord()实现将字符转换为 ASCII 码，运行结果如图 2.78 所示。

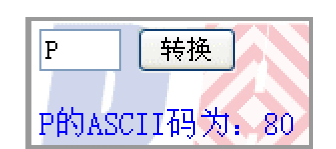


图 2.78　获取任意字符的 ASCII 码

关键技术

PHP 中获取字符的 ASCII 码应使用 ord()函数，其语法如下：

```
int ord ( string string);
```

该函数预期返回一个整型数值。

设计过程

（1）创建 PHP 脚本，当单击“转换”按钮时，通过字符串函数 ord()获取指定字符串的 ASCII 码，代码如下：

```
<?php
	if($_POST[sub]){                                              //通过 POST 方式传递参数
		echo "$_POST[text]的 ASCII 码为：";                  //输出
		echo ord($_POST[text]);                               //ASCII 转码
	}
?>
```

（2）将该文件存储于\MR\02\104 文件夹下，命名为 index.php。运行结果如图 2.78 所示。

秘笈心法

心法领悟 104：知识拓展。

像 HTML 标签“
”等都可以在 ASCII 码中找到相同功能的 ASCII 码值。

实例 105 查找指定 ASCII 码对应的字符

光盘位置：光盘\MR\02\105

高级

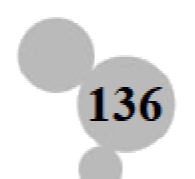
趣味指数：★★★★

实例说明

本实例通过字符串函数 chr()实现将 ASCII 码转换为字符，运行结果如图 2.79 所示。



图 2.79　查找指定 ASCII 码对应的字符



关键技术

将 ASCII 码转换为字符可以通过 chr()函数实现，其语法如下：

```
string chr ( int ascii);
```

该函数可将 ASCII 序数值转换成对应的字符，与 ord()函数成对照。

设计过程

（1）创建 PHP 脚本。当单击“转换”按钮时，首先，利用 POST 方法获取文本框中传入的数据信息，然后将数据信息利用 chr()函数转换为 ASCII 码并实现输出，代码如下：

```
<?php
	if($_POST[sub]){                              //通过 POST 方式传递参数
		echo "$_POST[text]的 ASCII 码为：";      //输出
		echo chr($_POST[text]);                   //ASCII 对应的字符
	}
?>
```

（2）将该文件存储于\MR\02\105 文件夹下，命名为 index.php。运行结果如图 2.79 所示。

秘笈心法

心法领悟 105：ASCII 码的优点。

ASCII 码在程序开发中的使用相当广泛，在某些时候将字符转换成 ASCII 码而被系统识别，既能保证安全性，又能获取更好的支持。

实例 106	合理定义上传文件的名称 光盘位置：光盘\MR\02\106	高级 趣味指数：★★★★

实例说明

为了使上传文件的名称不至于与服务器已有文件的名称重名，合理定义上传文件的名称就显得十分重要。本实例通过 rand()函数合理定义上传文件的名称，运行结果如图 2.80 所示。



图 2.80　合理定义上传文件的名称

关键技术

在 PHP 中应用 move_uploaded_file()函数实现文件的上传。但是在执行文件上传之前，为了防止潜在的攻击对原本不能通过脚本交互的文件进行非法管理，可以先应用 is_uploaded_file()函数判断指定的文件是不是通过 HTTP POST 上传的，如果是，则返回 TRUE。

（1）is_uploaded_file()函数

该函数用于判断指定的文件是不是通过 HTTP POST 上传的，其语法如下：

```
bool is_uploaded_file ( string filename )
```

参数 filename 必须指定类似于$_FILES['filename']['tmp_name']的变量，不可以使用从客户端上传的文件名$_FILES['filename']['name']。

通过 is_uploaded_file()函数对上传文件进行判断，可以确保恶意的用户无法欺骗脚本去访问本不能访问的文件，例如/etc/passwd。

（2）move_uploaded_file()函数

使用该函数将文件上传到服务器中指定的位置。如果成功返回 TRUE，否则返回 FALSE，语法如下：

```
bool move_uploaded_file ( string filename, string destination )
```

参数 filename 指定上传文件的临时文件名，即$_FILES[tmp_name]；参数 destination 指文件上传后保存的新路径和名称。如果参数 filename 不是合法的上传文件，不会出现任何操作，move_uploaded_file()将返回 FALSE；如果参数 filename 是合法的上传文件，但由于某些原因无法移动，也不会出现任何操作，move_uploaded_file()将返回 FALSE，此外还会发出一条警告。

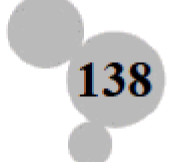
（3）rand()函数

使用该函数生成的随机整数作为上传文件的新名称。

设计过程

创建 index.php 文件，添加表单，设置文件域、提交按钮，使用 POST 方法，设置 enctype="multipart/form-data"，通过$_FILES 获取上传文件的相关信息，通过 move_upload_file()函数完成图片上传，其代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                    //判断上传内容是否为空
 if($_FILES['up_picture']['error']>0){                    //判断文件是否可以上传到服务器
        echo "上传错误:";
        switch($_FILES['up_picture']['error']){
                case 1:
                        echo "上传文件大小超出配置文件规定值";
                break;
                case 2:
                        echo "上传文件大小超出表单中约定值";
                break;
                case 3:
                        echo "上传文件不全";
                break;
                case 4:
                        echo "没有上传文件";
                break;
        }
}else{
        if(!is_dir("./upfile/")){                         //判断指定目录是否存在
                mkdir("./upfile/");                       //创建目录
        }
        $path='./upfile/'.rand().$_FILES['up_picture']['name'];              //定义文件名称和存储位置
        if(is_uploaded_file($_FILES['up_picture']['tmp_name'])){             //是不是 HTTP POST 上传
                if(!move_uploaded_file($_FILES['up_picture']['tmp_name'],$path)){   //执行上传
                        echo "上传失败";
                }else{
                        echo "文件".time().$_FILES['up_picture']['name']."上传成功，大小为：".$_FILES['up_picture']['size'];
                }
        }else{
                echo "上传文件".$_FILES['up_pictute']['name']."不合法！";
        }
}
}
?>
```



秘笈心法

心法领悟 106：上传文件的命名问题。

上传文件如果和已有文件重名，文件可能会相互替换，给网站造成损失。所以确保上传文件名称的唯一性是非常必要的，也可以通过 time()时间戳来定义上传文件的名称。

实例 107	通过异或方式对用户注册密码进行加密 光盘位置：光盘\MR\02\107	高级 趣味指数：★★★★

实例说明

本实例通过异或方式对用户注册的密码进行加密，并且输出加密后的密码，运行结果如图 2.81 所示。

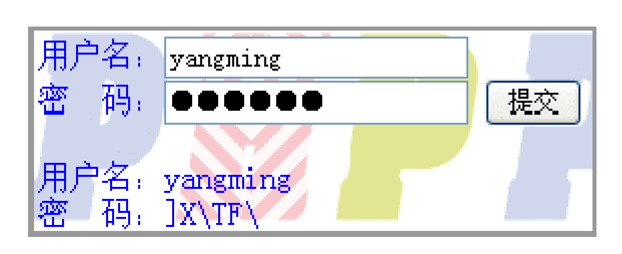


图 2.81　通过异或方式对用户注册密码进行加密

关键技术

通过异或方式对字符串加密的原理：当一个字符串 A 与另一个字符串 B 进行异或运算后会产生一个字符串 C。

设计过程

（1）创建 PHP 脚本。首先，定义字符串变量并为此字符串变量赋值。然后，通过异或方式对字符串变量进行加密并输出加密结果，代码如下：

```
<?php
	if($_POST[sub]){	//通过 POST 方式传递参数
		$str = "0123456798";	//定义字符串
		echo "用户名：".$_POST[text]."<br>";
		echo "密  码：".($_POST[pwd]^$str);	//异或加密
	}
?>
```

（2）将该文件存储于\MR\02\107 文件夹中，命名为 index.php。运行结果如图 2.81 所示。

秘笈心法

心法领悟 107：比较异或方式与 md5 加密技术。

PHP 中对字符串加密的方式有很多种，如 md5 加密等。异或方式与 md5 加密方式的本质区别是 md5 加密后不能解密，异或方式可以。相对而言，md5 加密更加安全。

实例 108	统一英文注册用户首字母的大小写 光盘位置：光盘\MR\02\108	高级 趣味指数：★★★★

实例说明

网站的国际化已经不是一个炙手可热的话题，但是由于地区差异，在使用英文填写注册信息时，首字母的

大小写往往得不到统一。本实例利用 ucfirst()函数实现将英文注册用户的首字母统一为大写，运行结果如图 2.82 所示。



图 2.82　统一英文注册用户首字母的大小写

关键技术

在 PHP 开发中，定义首字母大写的方法有很多，但是所有的方法都没有使用 PHP 预定义函数 ucfirst()简单。ucfirst()函数的语法如下：

```
string ucfirst ( string str );
```

参数说明：

string str：定义字符串。

设计过程

（1）创建 PHP 脚本。当单击“提交”按钮时，首先利用 POST()方法获取从文本框传递进来的数据，然后将获取的数据经 ucfirst()函数转换首字母大写，最后输出经转换后的数据，代码如下：

```
<?php
	if($_POST[sub]){                                    //通过 POST 方式传递参数
		$a = $_POST[text];                              //接收参数
		echo "首字母统一为大写:";
		echo ucfirst($a);                               //转换单词首字母
	}
?>
```

（2）将该文件存储于\MR\02\108 文件夹中，命名为 index.php。运行结果如图 2.82 所示。

秘笈心法

心法领悟 108：知识拓展。

实现字符串首字母大小写转换还可以通过转码来完成。首先利用 substr()函数提取出字符串的首字母，之后利用 ord()函数获取该字母的 ASCII 码，再根据大小写字母之间相差 32 来实现大小写字母之间的 ASCII 码转换，最后利用 chr()函数将转变后的 ASCII 码转变为字母即可。

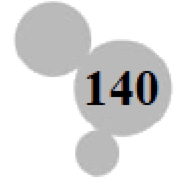
实例 109	字符串与数字之间的转换 光盘位置：光盘\MR\02\109	高级 趣味指数：★★★★

实例说明

来看两个变量$a = 123;和$a = '123';。前者是整型变量，后者是字符型变量。本实例利用函数使字符型变量转换为整型变量，整型变量转换为字符型变量，运行结果如图 2.83 所示。

转换前类型为：string，原字符串输出：123
转换后输出：123，转换后类型为：integer
转换前类型为：integer，原字符串输出：123
转换后输出：123，转换后类型为：string

图 2.83　字符串与数字之间的转换



关键技术

在 PHP 中，将数字转换为字符串可以通过 strval()函数实现，其语法如下：

```
string strval ( mixed var)
```

该函数用于返回变量 var 对应的 string 值。

设计过程

（1）首先创建脚本文件，定义字符串变量并将数值型字符串赋给变量，然后通过 strval()函数转换数据，代码如下：

```
<?php
	$str = "123";                              //定义字符串
	echo "转换前类型为：".gettype($str);          //取得当前数据类型
	echo "，原字符串输出：".$str;
	echo "<br>";
	echo "转换后输出：".intval($str);             //将字符串转换为整型
	echo "，转换后类型为：".gettype(intval($str)); //取得转换后的数据类型
	echo "<hr>";
	$int = 123;                                //定义整型变量
	echo "转换前类型为：".gettype($int);          //取得当前类型
	echo "，原字符串输出：".$int;
	echo "<br>";
	echo "转换后输出：".strval($int);             //转换为字符串
	echo "，转换后类型为：".gettype(strval($int)); //输出转换后的数据类型
?>
```

（2）将该文件存储于\MR\02\109 文件夹下，命名为 index.php。运行结果如图 2.83 所示。

秘笈心法

心法领悟 109：简述 strval()和 intval()函数。

strval()和 intval()函数在实现功能上与强制类型转换机制相同。

实例 110　通过正则表达式对字符串进行匹配查找

光盘位置：光盘\MR\02\110

高级

趣味指数：★★★★

实例说明

字符串匹配查找的方法有很多，下面来了解如何用正则表达式对字符串进行匹配查找。本实例通过数组函数和正则表达式函数 preg_match()来实现字符串的匹配查找，运行结果如图 2.84 所示。



图 2.84　利用正则表达式实现字符串的匹配查找

关键技术

灵活运用正则表达式函数是本实例的关键。preg_match()函数用于匹配指定字符串，一般情况下，会与 if 条件语句协同使用，其语法如下：

```
int preg_match ( string pattern, string subject [, array matches] )
```

函数功能：在字符串 subject 中匹配表达式 pattern。函数返回匹配的次数。如果有数组 matches，那么每次

匹配的结果将被存储到数组 matches 中。该函数在匹配成功后就停止继续查找，其返回值是 0 或 1。

设计过程

（1）创建 PHP 脚本。首先，定义数组变量$a 并为$a 赋值。然后，利用 foreach 语句遍历数组以便取得 key 值和 value 值。最后，利用正则表达式函数 preg_match()将通过 POST 方法传递进来的数据进行匹配，匹配成功则输出对应数组的 value 值，代码如下：

```
<?php
	$a = array('mrsoft','明日 mr 科技','www.mrsoft.com');				//定义数组
	foreach($a as $key => $value){								//foreach 循环遍历数组
		$a = $_POST[te];
		if(preg_match("/$a/",$value)){						//正则表达式匹配
			echo $value;
			echo "<br>";
		}
	}
?>
```

（2）将该文件存储于\MR\02\110 文件夹下，命名为 index.php。运行结果如图 2.84 所示。

秘笈心法

心法领悟 110：SQL 语句的匹配问题。

PHP 项目开发过程中可以利用 SQL 语句中的关键字 like，对数据库中的内容进行匹配查找，其实现的是数据库中数据的模糊查询。

实例 111 通过 IP 地址查找主机所在地

光盘位置：光盘\MR\02\111

高级

趣味指数：★★★★

实例说明

IP 地址是互联网身份的象征，通过 IP 地址可以得到用户的相关信息。本实例通过正则表达式验证 IP 地址是否合法，然后实现 IP 地址的格式转换，进而获取 IP 地址所在地，运行结果如图 2.85 所示。

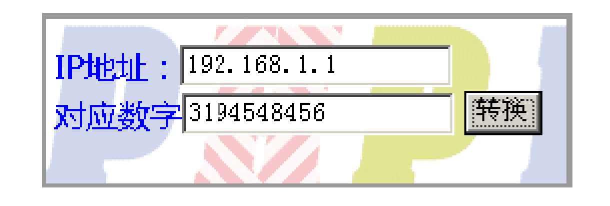


图 2.85　利用 IP 地址查找主机所在地

关键技术

首先需要验证输入的 IP 地址是否合法，如果合法，再将其转换为对应的数值。验证 IP 地址是否合法可以通过正则表达式实现。

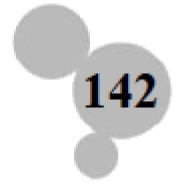
IP 地址使用 4 位地址定位设备，它采用点分十进制数的格式“xxx.xxx.xxx.xxx”，例如 192.168.1.66。因此，验证 IP 地址是否合法的正则表达式如下：

```
/(\d+).(\d+).(\d+).(\d+)/
```

将 IP 地址转换为对应的数值可以应用以下公式实现：

```
P1*255*255*255+P2*255*255+P3*255+P4*1
```

在上面的公式中，P1 代表 IP 地址第 1 段的数值；P2 代表 IP 地址第 2 段的数值；P3 代表 IP 地址第 3 段的数值；P4 代表 IP 地址第 4 段的数值。



设计过程

（1）编写 JavaScript 脚本，定义 deal()方法验证用户提交的 IP 地址格式是否正确，定义 convert()方法将 IP 地址转换为对应数值，该方法只有一个参数 ip，用于指定 IP 地址，返回值为 IP 地址对应的数值，代码如下：

```
<script language="javascript">
	function deal(myform){                                          //自定义函数
		var ip=form1.ip.value;                                      //定义变量
		objExp=/(\d+).(\d+).(\d+).(\d+)/;                           //正则表达式
		if(ip==""){                                                 //条件判断
			alert("请输入 IP 地址!");form1.ip.focus();return;
	}
	if(objExp.test(ip)){
			myform.ipNum.value=convert(ip);
		}else{
			alert("您输入的 IP 地址不合法!");form1.ip.focus();return;
		}
}
	function convert(ip){                                           //自定义函数
		ip=ip.split(".");
		vip=ip[0]*255*255*255+ip[1]*255*255+ip[2]*255+ip[3]*1;     //转换程式
		return vip;                                                 //返回值
	}
</script>
```

（2）创建 index.php 文件，通过 form 表单提交要验证的 IP 地址，调用 JavaScript 脚本中的方法验证 IP 地址格式并完成格式的转换操作。

秘笈心法

心法领悟 111：使用$_SERVER["REMOTE_ADDR"]的注意事项。

在 PHP 中，可以通过$_SERVER["REMOTE_ADDR"]获取客户机的 IP 地址，但并不能判断该 IP 地址的所在地。如果要判断 IP 地址的所在地，就必须对其格式进行转换。

实例 112　解决用 substr()函数对中文字符串截取时出现乱码的问题

光盘位置：光盘\MR\02\112

高级

趣味指数：★★★★

实例说明

使用 substr()函数是按字节截取字符串，在截取中文字符串时，由于一个汉字由两个字符组成，如果只截取 1 个字符就会出现乱码。本实例使用自定义函数解决对中文字符串截取时的乱码问题，运行结果如图 2.86 所示。

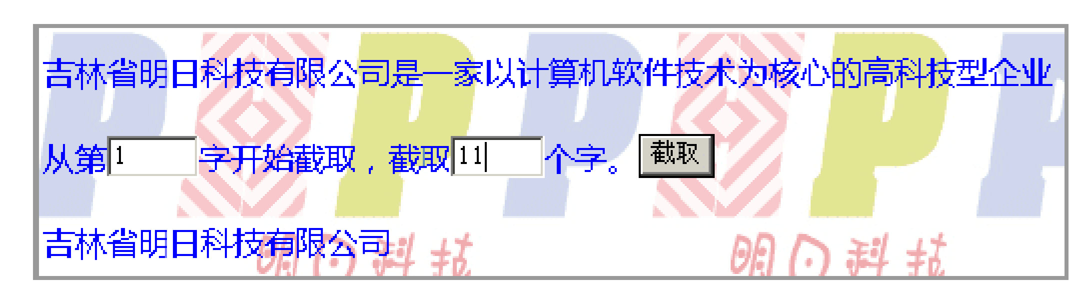


图 2.86　解决用 substr()函数对中文字符串截取时的乱码

关键技术

对中文字符串的截取虽说是通过自定义函数来完成的，但是其根本还是应用 substr()函数，只是在进行字符串截取时，对字符串的类型进行了判断。

对截取字符串中首个字节的 ASCII 序数值进行判断，如果 ASCII 序数值大于 0xa0，则表示为汉字，那么在

应用 substr()函数进行截取时，就以 2 个字节为单位；如果 ASCII 序数值小于 0xa0，则表示为英文字符串，那么在应用 substr()函数进行截取时，就以 1 个字节为单位。这样将中文字符串和英文字符串分隔进行截取就避免了出现乱码的问题。

自定义函数 msubstr()的语法如下：

```
function msubstr($str, $start, $len) {                //$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
 $strlen = $start + $len;                             //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
 for($i = 0; $i < $strlen; $i ++) {                   //通过 for 循环语句，循环读取字符串
        if (ord ( substr ( $str, $i, 1 ) ) > 0xa0) {  //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
               $tmpstr .= substr ( $str, $i, 2 );     //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
               $i ++; //变量自加 1
        } else {                                      //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
               $tmpstr .= substr ( $str, $i, 1 );
        }
 }
 return $tmpstr;                                      //输出字符串
}
```

其中，参数$str 是指定被截取的字符串；参数$start 是截取的开始位置；参数$len 是截取的长度。返回值$tmpstr 是截取后的字符串。

设计过程

创建 index.php 文件，定义 msubstr()函数，对中文字符串进行截取。创建 form 表单，提交字符串截取的开始位置和长度，然后调用 msubstr()函数对字符串进行截取并且输出截取结果，代码如下：

```
<?php
function msubstr($str, $start, $len) {                //$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
 $strlen = $start + $len;                             //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
 for($i = 0; $i < $strlen; $i ++) {                   //通过 for 循环语句，循环读取字符串
        if (ord ( substr ( $str, $i, 1 ) ) > 0xa0) {  //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
               $tmpstr .= substr ( $str, $i, 2 );     //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
               $i ++;                                 //变量自加 1
        } else {                                      //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
               $tmpstr .= substr ( $str, $i, 1 );
        }
 }
 return $tmpstr;                                      //输出字符串
}
$string="吉林省明日科技有限公司是一家以计算机软件技术为核心的高科技型企业";
$str=msubstr($string,$_POST[te],$_POST[tx]);
echo $str;
?>
```

秘笈心法

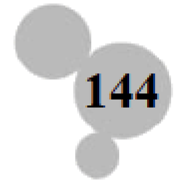
心法领悟 112：利用函数截取中文字符串的注意事项。

使用 substr()函数截取中文字符串时最好首先定义算法，以免截取半个中文文字而出现乱码。

实例 113	字符串与 HTML 标记相互转换 光盘位置：光盘\MR\02\113	高级 趣味指数：★★★★

实例说明

本实例是通过 PHP 5.0 中的新型字符串实现字符串与 HTML 标记的相互转换，运行结果如图 2.87 所示。



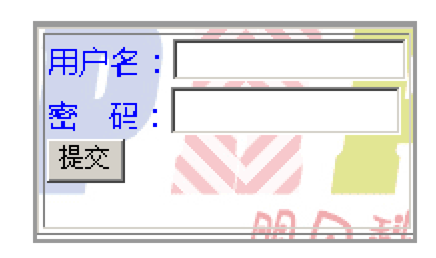


图 2.87　字符串与 HTML 标记的相互转换

关键技术

本实例的关键点是 PHP 5.0 新型字符串的使用，详细说明请参见实例 038。

设计过程

创建 index.php 文件，利用 PHP 5.0 新型字符串将 HTML 标记封装到 PHP 的变量中，并且输出 PHP 变量的值，代码如下：

```
<?php
$str=<<<mark                                        //新型字符串格式
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>字符串与 HTML 标记相互转换</title>
</head>

<body background="../image/1.jpg" style="font-size:16px; color:blue; font-family:'微软雅黑'"><table border="1"><tr><td>
<form action="" method="post">
        用户名：<input type="text" name="test" size="15" /><br>
        密   码：<input type="password" name="pwd" size="15" /><br>
        <input type="submit" name="sub" value="提交" />
</form></td></tr></table>
</body>
</html>
mark;                                               //新型字符串结束符
echo $str;
?>
```

秘笈心法

心法领悟 113：新型字符串的应用范围。

新型字符串被广泛应用在含有较多 HTML 标记的 PHP 编码中。

实例 114　运用 PHP 5.0 新型字符串输出 XML 数据

光盘位置：光盘\MR\02\114

高级

趣味指数：★★★★

实例说明

本实例通过 PHP 5.0 新型字符串输出 XML 格式的数据，运行结果如图 2.88 所示。



图 2.88　PHP 5.0 新型字符串输出 XML 格式的数据

关键技术

本实例的关键点是 PHP 5.0 新型字符串的使用，详细说明请参见实例 038。

设计过程

创建 index.php 文件，定义 PHP 5.0 新型字符串，通过其输出 XML 格式的数据，代码如下：

```
<?php
$str=<<<mark
<?xml version="1.0" encoding="utf8"?>
<rss xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:taxo="http://purl.org/rss/1.0/modules/taxonomy/" version="2.0">
  <channel>
    <title>明日科技</title>
    <link>http://www.mingrisoft.com</link>
    <description>吉林省明日科技有限公司</description>
    <dc:creator>http://www.mingrisoft.com</dc:creator>
<item>
      <title>编程词典</title>
      <link>http://www.mrbccd.com</link>
      <description>明日科技编程词典重拳出击，隆重上市！</description>
      <pubDate>2008-08-08</pubDate>
      </item>
</channel></rss>
mark;
echo $str;
?>
```

秘笈心法

心法领悟 114：使用新型字符串的优点。

在 PHP 5.0 之前的开发中，想要在 PHP 中输出一些 HTML 标记是通过 echo 语句一行一行输出的，这无疑是一件非常头痛的事。现在 PHP 提供了新型字符串，相关的问题就可迎刃而解了。

实例 115 判断字符串中是否存在指定子串

光盘位置：光盘\MR\02\115

高级

趣味指数：★★★★

实例说明

判断一个字符串中是否含有另一个字符串的方法有很多，例如，正则表达式等。本实例通过 stristr()函数判断字符串中是否存在指定子串，运行结果如图 2.89 所示。

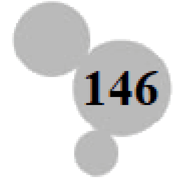


图 2.89　字符串中不存在子串

关键技术

判断一个字符串是不是另一个字符串的子串可通过 stristr()函数实现，其语法如下：

```
string stristr ( string haystack, string needle );
```



该函数获取指定字符串（A）在另一个字符串（B）中首次出现的位置到字符串（B）末尾的所有字符串。参数 haystack 指定查找的字符串，参数 needle 指定查找的对象。

该函数如果执行成功则返回剩余的字符串，否则将返回 FALSE。

设计过程

创建 index.php 文件，定义字符串变量，将该变量与通过 POST()方法传递进来的数据利用 stristr()函数进行比较并输出结果，代码如下：

```
<?php
    $a = "Hello World !!";                                          //定义字符串
    echo $a;                                                        //输出字符串
    echo "<form action="method='post'>";                            //输出表单
    echo "<input type='text' name='text' value='输入文本内容'>";
    echo "<input type='submit' name='sub' value='提交'>";
    echo "</form>";
    if($_POST[sub]){                                                //通过 POST 方式传递参数
        if(stristr($a, $_POST[text]) != ""){                        //判断是否存在子串
            echo "<script>alert('文本存在指定子串');</script>";      //提示
        }else{
            echo   "<script>alert('文本不存在指定子串');</script>";  //提示
        }
    }
?>
```

秘笈心法

心法领悟 115：知识扩展。

stristr()函数还可以作为分割函数的一部分，从符合条件的字符串开始截取，一直到文件末尾。该函数不区分大小写，如果要在区分字母大小写的情况下进行搜索，可以应用 strstr()函数。

2.9　正则表达式

正则表达式是一种描述字符串结构模式的形式化表达方法，是一个强大、便捷、高效的文本处理工具，可用于验证用户输入的数据和检索大量的文本。在 PHP 中，表达式应用最好的体现是对表单提交的数据进行验证，判断是否合理、合法。

实例 116	验证电话号码的格式是否正确 光盘位置：光盘\MR\02\116	高级 趣味指数：★★★★

实例说明

表单注册时往往要求用户书写座机电话号码，而座机电话号码是由 11 位或 12 位数字组成的，所以一定要对电话号码的位数和格式进行限制。本实例通过正则表达式和正则表达式函数 preg_match()实现对电话号码格式的验证，运行结果如图 2.90 所示。



图 2.90　验证电话号码的格式是否正确

关键技术

（1）preg_match()：进行正则表达式匹配。

```
int preg_match ( string pattern, string subject [, array matches [, int flags]]);
```

（2）正则表达式：固定总位数为 11 位或 12 位的座机电话。

```
/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/;
```

设计过程

创建 index.php 文件，通过正则表达式函数对输入文本框的电话号码进行验证，并输出相关提示，其代码如下：

```
<?php
    if($_POST['sub']){
        if(preg_match("/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/",$_POST['text'])){          //正则表达式验证
            include("inc.php");                                                          //包含信息提示页
            show_error("信息提示","格式正确","index.php");                              //调用自定义函数输出提示信息
        }else{
            include("inc.php");
            show_error("信息提示","格式错误","index.php");                              //调用自定义函数输出提示信息
        }
    }
?>
```

秘笈心法

心法领悟 116：使用正则表达式的注意事项。

正则表达式“/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/”是根据新式电话号码设置的，老式的座机电话是由 10 位数字组成的。

实例 117	验证 E-mail 地址格式是否正确 光盘位置：光盘\MR\02\117	高级 趣味指数：★★★★

实例说明

互联网发展到今天，几乎所有的 Web 爱好者都有自己的 E-mail 地址，无论申请的是 126 邮箱还是 163 邮箱，E-mail 地址的格式是固定的。本实例通过 preg_match()正则匹配函数和正则表达式验证 E-mail 地址格式是否正确，运行结果如图 2.91 所示。



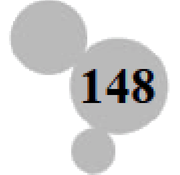
图 2.91　验证 E-mail 地址格式是否正确

关键技术

（1）preg_match()函数：进行正则表达式匹配，详细参数参见实例 110。

（2）正则表达式：对 E-mail 地址进行验证。

```
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;
```



设计过程

创建 index.php 文件，当单击“验证”按钮时，通过正则表达式函数验证文本框传入的信息并输出对应提示，其代码如下：

```
if($_POST['sub']){
        require_once("inc.php");                                                        //包含信息提示页
        if(preg_match("/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/",trim($_POST['text']))){    //正则表达式验证
                show_error("信息提示","格式正确","index.php");                          //输出提示
        }else{
                show_error("信息提示","格式不正确","index.php");                        //输出提示
        }
}
?>
```

秘笈心法

心法领悟 117：PHP 支持的两种正则表达式函数库。

PHP 支持两种正则表达式函数库，一种是正则表达式函数库（POSIX 扩展），另外一种是正则表达式函数库（Perl 兼容）。在性能上，Perl 兼容正则表达式速度更快一些。

实例 118	验证 IP 地址是否有效 光盘位置：光盘\MR\02\118	高级 趣味指数：★★★★

实例说明

IP 地址是 Web 用户可以访问互联网的身份凭证。每一个 IP 地址相对其他用户的 IP 都是独立的。本实例通过正则表达式函数 preg_match()和正则表达式对 IP 地址进行验证，运行结果如图 2.92 所示。



图 2.92　验证 IP 地址是否有效

关键技术

（1）preg_match_all()函数，在字符串 subject 中匹配表达式 pattern，函数返回匹配的次数。如果有数组 matches，那么每次匹配的结果将被存储到数组 matches 中，其语法如下：

```
int preg_match_all ( string pattern, string subject [, array matches] )
```

（2）正则表达式：对 IP 地址进行验证。

```
"/\d+\\d+\\d+\.\d+/";
```

设计过程

（1）创建 index.php 文件，通过 form 表单提交 IP 地址，当单击“验证”按钮时，通过正则表达式函数验证 IP 地址，并且返回验证结果，其代码如下：

```
<?php
if($_POST['sub']){
```

```
        require_once("inc.php");
        if(preg_match("/\d+\\d+\\d+\.\d+/",trim($_POST['text']))){
                show_error("信息提示","格式正确","index.php");
        }else{
                show_error("信息提示","格式不正确","index.php");
        }
}
?>
```

（2）创建 inc.php 文件，定义方法 show_error()，根据传递的参数值定义返回的错误信息。

秘笈心法

心法领悟 118：IP 地址是什么。

Internet 为每一个主机分配唯一的一个 32 位地址，该地址称为 IP 地址，也称网际地址。IP 地址由 4 个数组成，每个数取值范围为 0~255，每两个数之间用“.”分隔。所以 IP 地址的格式是固定的，正则表达式“/\d+\\d+\\d+\.\d+/”可以验证所有 IP 地址。

实例 119　统计关键字的查询结果

光盘位置：光盘\MR\02\119

高级

趣味指数：★★★★

实例说明

统计关键字的查询结果的方法有很多，本实例通过正则表达式函数 split()和 count()实现统计关键字，运行结果如图 2.93 所示。

PHP作为全球最普及、应用最广泛的互联网开发语言之一，从1994年诞生至今已被2000多万个动态网站采用，全球知名互联网公司Google、Yahoo、eBay和中国知名网站新浪、百度、阿里巴巴等均采用PHP技术！

输出关键字P一共出现4次

图 2.93　统计关键字的查询结果

关键技术

split()函数：用正则表达式将字符串分割到数组中。

```
array split ( string pattern, string string [, int limit]);
```

该函数返回一个字符串数组，每个单元为 string 经区分大小写的正则表达式 pattern 作为边界分割出的子串。如果设定了 limit，则返回的数组最多包含 limit 个单元，而其中最后一个单元包含 string 中剩余的所有部分。如果出错，则 split()函数返回 FALSE。

设计过程

创建 index.php 文件。首先，定义字符串变量$str 并为$str 赋值。然后，利用 split()函数检索字符 p 在$str 中的出现次数，并输出检索的结果，其代码如下：

```
<?php
//定义字符串变量
$str = "PHP 作为全球最普及、应用最广泛的互联网开发语言之一，从 1994 年诞生至今已被 2000 多万个动态网站采用，全球知名互联网公司
Google、Yahoo、eBay 和中国知名网站新浪、百度、阿里巴巴等均采用 PHP 技术！ ";
$b = split("P",$str);                                          //使用 split()函数进行分割
echo "<h4 style ='red'>";
        echo $str."<br>";
        echo "</h4>";
```



```
	echo "输出关键字 P 一共出现".(count($b)-1)."次";			//利用 count()函数进行统计
?>
```

秘笈心法

心法领悟 119：split()函数的实现原理。

split()正则表达式函数是将字符串分割到数组中，如果字符串中有 n 个与 pattern 匹配的项目，则返回的数组将包含 n+1 个单元，所以 count()在计算时要减 1。

实例 120　计算查询操作的执行时间

光盘位置：光盘\MR\02\120

高级

趣味指数：★★★★

实例说明

计算机的每一步操作都是需要时间的，只不过由于时间过于短暂，用户感觉不到。本实例通过时间戳函数 microtime()计算查询操作的执行时间，运行结果如图 2.94 所示。



图 2.94　计算查询操作的执行时间

关键技术

获取查询操作的执行时间，在 PHP 中主要是使用 microtime()函数实现，其语法如下：

```
mixed microtime ( [bool get_as_float] );
```

该函数获取某一时刻的时间戳的微秒数。

设计过程

创建 index.php 文件。首先，连接数据库并设置页面的编码风格，在执行查询操作的开始与结束的两个时间点定义两个变量，分别利用 microtime()函数获取两点时间戳的微秒数。然后计算两个时间戳的差值，取得程序的执行时间，其代码如下：

```
<?php
	$conn = mysql_connect("localhost","root","111")or die("连接 mysql 出现错误");		//连接 MySQL
	mysql_select_db("db_database02",$conn);						//连接数据库
	$sql = "SELECT * FROM tb_select";						//SQL 语句
	$a = microtime();								//取得当前时间戳的微秒数
	$rs = mysql_query($sql);							//查询操作
	while($rst = mysql_fetch_array($rs))
	{
		echo "id:".$rst[0]."<br>";
		}
	$b = microtime();								//取得当前时间戳的微秒数
	echo "查询操作所用时间为：".($b-$a)."秒";					//输出结果
?>
```

说明：本实例所用的数据表没有足够多的数据，以至于执行速度太快几乎不需要时间，为了显示结果，笔者人为让程序休眠 1 秒钟。

秘笈心法

心法领悟 120：知识扩展。

time()和 mktime()函数在不指定参数的情况下具有相同的作用，都是返回自从 UNIX 新纪元（格林威治时间 1970 年 1 月 1 日 00:00:00）到当前时间的秒数，而 microtime()函数只是单纯取得两个时间戳的微秒数。

实例 121 通过正则表达式替换函数查询关键字描红

光盘位置：光盘\MR\02\121

高级

趣味指数：★★★★

实例说明

关键字描红技术在实例 088 中已经进行过讲解，本实例是通过正则表达式替换函数 preg_replace()实现查询关键字描红，运行结果如图 2.95 所示。

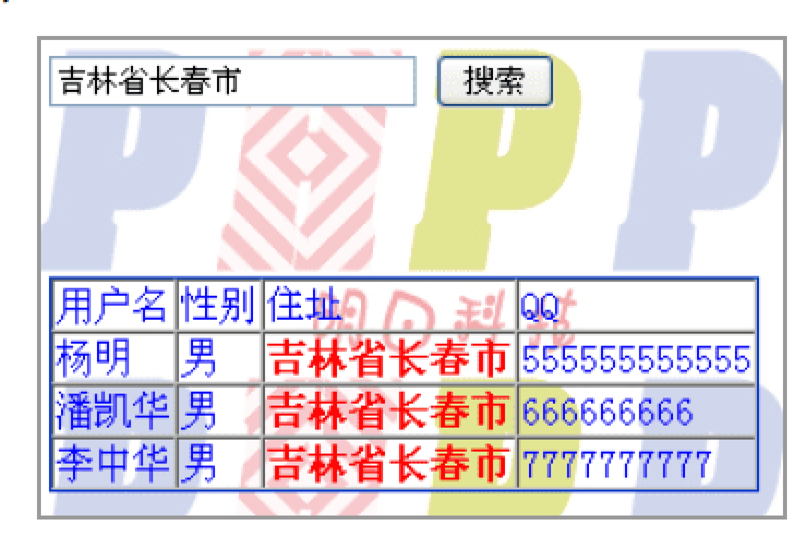


图 2.95　查询关键字描红

关键技术

执行替换或关键字描红时经常使用 PHP 预定义函数 preg_replace()，其语法如下：

```
mixed preg_replace ( mixed pattern, mixed replacement, mixed subject [, int limit] );
```

参数说明：

mixed pattern：规定替换规则。

mixed replacement：替换部分。

mixed subject：替换内容的源信息。

主要功能：执行正则表达式的搜索和替换。

设计过程

创建 index.php 文件。首先连接数据库并设置页面的编码风格，当单击“搜索”按钮时，执行查询操作并将文本框输入的信息进行描红处理，最后利用正则表达式函数 preg_replace()替换数据信息，其代码如下：

```
<?php
	$conn = mysql_connect("localhost","root","111")or die("连接 mysql 出现错误");		//连接 MySQL
	mysql_select_db("db_database02",$conn);		//连接数据库
	mysql_query("SET NAMES GBK");		//设置编码格式
	if($_POST[sub]){		//通过 POST 方式传递参数
		//SQL 语句
		$sql = "select * from tb_sel where tb_sel.name like ('%$_POST[text]%') or tb_sel.add like ('%$_POST[text]%') or tb_sel.sex like
('%$_POST[text]%') or tb_sel.QQ like('%$_POST[text]%')";
		$rs = mysql_query($sql);
		$a = $_POST[text];		//原参数
		$b = "<b><font color='#FF0000'>".$_POST[text]."</font></b>";		//描红后的参数
?>
	<table border="1" bordercolor="#0033CC" cellspacing="0"><tr><td>用户名</td><td>性别</td><td>住址</td><td>QQ</td></tr>
<?php
		while($rst = mysql_fetch_array($rs)){		//查询输出
			$rste = preg_replace("/$a/",$b,$rst);		//正则替换
```

```
                                                            //输出表格
            echo "<tr><td>".$rste[name]."</td><td>".$rste[sex]."</td><td>".$rste[add]."</td><td>".$rste[QQ]."</td></tr><br>";
        }
    }
?>
</table>
```

秘笈心法

心法领悟 121：使用 preg_replace()函数的相关说明。

preg_replace()函数的每个参数（除了 limit）都可以是一个数组。如果 pattern 和 replacement 都是数组，将以其键名在数组中出现的顺序来进行处理。

实例 122　判断上传文件的类型

光盘位置：光盘\MR\02\122

高级

趣味指数：★★★★

实例说明

前面的实例中已经介绍了对文件类型的相关操作，本实例采用正则表达式函数 preg_match()实现判断上传文件的类型，运行结果如图 2.96 所示。

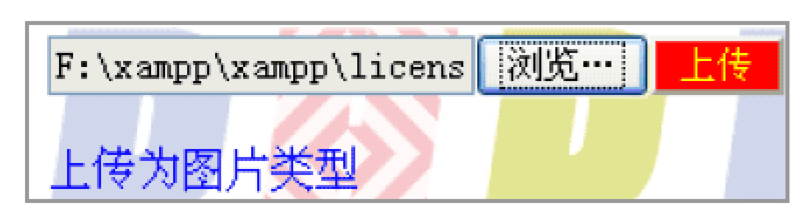


图 2.96　判断上传文件的类型

关键技术

使用 preg_match()函数进行正则表达式匹配，详细说明参见实例 110。

设计过程

创建 index.php 文件，当单击“上传”按钮时，利用 preg_match()正则表达式函数将上传的数据信息进行匹配并输出结果，其代码如下：

```
<?php
    if($_POST[sub]){                                        // if 条件语句
        if(preg_match("/.jpg/",strtolower($_POST[text]))){  //通过正则表达式函数判断类型
            echo "上传为图片类型";
        }else{
            if(preg_match("/.rar/",strtolower($_POST[text]))){
                echo "上传为压缩包类型";
            }else{
                echo "其他文件类型";
            }
        }
    }
?>
```

秘笈心法

心法领悟 122：知识扩展。

本实例使用的是 if…else…语句嵌套，从结构上看不是很明朗，读者可以运用 switch 语句改写此实例，相信在系统运行效率上会有所提高。

实例 123 判断邮政编码格式是否正确

光盘位置：光盘\MR\02\123

高级

趣味指数：★★★★

实例说明

用户在购买电子商务网站的商品时，商品一般都是以快递或邮寄的方式传递到用户手中，所以如果用户不能正确地填写邮寄地址或邮政编码，就有可能造成不必要的损失。本实例通过正则表达式函数 preg_match()验证用户提交的邮政编码格式是否正确，运行结果如图 2.97 所示。



图 2.97 验证邮政编码是否有效

关键技术

（1）preg_match()函数：进行正则表达式匹配，详细参数参见实例 110。

（2）正则表达式：对邮政编码格式进行验证。

```
"/[0-9]{6}/";
```

设计过程

创建 index.php 文件，当单击"验证"按钮时，利用 preg_match()正则表达式函数对文本框的输入信息进行匹配并输出提示，其代码如下：

```
<?php
	if($_POST['sub']){								//通过 POST 方式传递参数
		require_once("inc.php");						//包含文件
		if(preg_match("/[0-9]{6}/",trim($_POST['text']))){		//正则表达式
			show_error("信息提示","格式正确","index.php");		//显示信息
		}else{
			show_error("信息提示","格式不正确","index.php");
		}
	}
?>
```

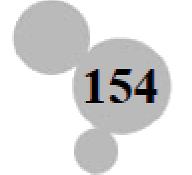
秘笈心法

心法领悟 123：什么是邮政编码。

邮政编码是为了实现邮政分拣自动化和邮政网络数字化设置的，以加快邮件邮递速度。目前世界上有 40 多个国家先后实行了邮政编码制度，并以此作为衡量一个国家通信技术和邮政服务水平的标准之一。各国邮政编码规则并不统一。

2.10 数　组

数组是对大量数据进行组织和管理的有效手段之一，通过数组的强大功能，可以对大量性质相同的数据进



行存储、排序、插入及删除等操作，从而可以有效地提高程序开发效率及改善程序的编写方式。

实例 124　**创建并输出数组**　高级

光盘位置：光盘\MR\02\124　趣味指数：★★★★

实例说明

如果只是想保存几个数据就没有必要和数据库交互，此时可以选用数组。本实例通过数组函数 array()实现创建并输出数组，运行结果如图 2.98 所示。

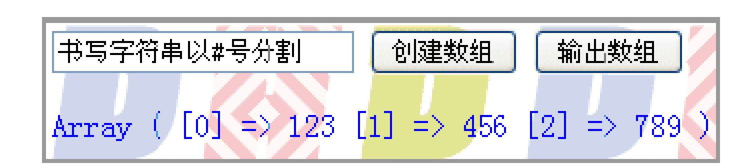


图 2.98　输出数组

关键技术

array()：新建一个数组。

```
array array ( [mixed ...] );
```

参数 mixed 的语法为“key=>value”，多个 mixed 参数之间用逗号隔开，分别定义了索引和值。索引可以是字符串或数字。如果省略了索引将是目前最大的整数索引+1。如果定义了两个完全一样的索引，则后面一个会覆盖前一个。数组中的各数据元素的数据类型可以不同，也可以是数组类型。当 mixed 是数组类型时，该数组就是二维数组。

设计过程

创建 index.php 文件，当单击“创建数组”按钮时，首先，判断文本框信息是否为空，如果为空则输出提示，否则将文本框信息保存在 SESSION 变量中；当单击“输出数组”按钮时，首先，判断网页中是否存在 SESSION 变量，如果存在变量则利用字符串拆分函数进行拆分并打印输出，其代码如下：

```
<?php
	if($_POST[sub1]){                                                              //通过 POST 方式传递参数
		if($_POST[text] == ""){                                                    //如果文本框为空
			echo "<script>alert('您创建了一个空数组');location.href='index.php'</script>";   //输出提示
		}else{
			$_SESSION['array'] = $_POST[text];                                     //将结果保存在 SESSION 中
			echo "<script>alert('创建数组成功');location.href='index.php'</script>";        //提示
		}
	}
	if($_POST[sub2]){                                                              //通过 POST 方式传递参数
		if(isset($_SESSION['array'])){                                             //如果 SESSION 存在
			$array = explode("#",$_SESSION['array']);                              //分割字符串
			print_r($array);                                                       //打印
			session_destroy();                                                     //销毁 SESSION
		}else{
			echo "<script>alert('请先创建数组');location.href='index.php'</script>";        //提示
		}
	}
?>
```

秘笈心法

心法领悟 124：数组的重要作用。

数组是 PHP 基础语言中很重要的一部分，灵活运用数组会起到事半功倍的效果。

实例 125 统计数组元素个数

光盘位置：光盘\MR\02\125

高级

趣味指数：★★★★

实例说明

数组下标默认是以 0 为开始键值的整型数据，如果想取得数组元素的个数就要使用 count()函数。本实例通过 count()函数统计数组元素个数，运行结果如图 2.99 所示。



图 2.99 统计数组元素个数

关键技术

在 PHP 中，应用 count()函数对数组中的元素个数进行统计，其语法如下：

```
int count ( mixed array [, int mode])
```

参数说明：

array：必选参数，指定被统计的数组。

mode：可选参数，其参数值为 COUNT_RECURSIVE（或 1），如选中此参数，本函数将递归地对数组计数。对计算多维数组的所有单元尤其有用。此参数的默认值是 0。

设计过程

创建 index.php 文件，首先定义数组变量并为此变量赋值，然后利用 count()函数输出数组元素的个数，其代码如下：

```
<?php
$a = array('1','精细','asd','asd123');              //声明数组
echo "数组中存在元素".count($a)."个";              //计算数组元素个数
?>
```

秘笈心法

心法领悟 125：使用 count()函数的注意事项。

如果 count()函数的操作对象是 NULL，那么返回结果为 0。count()函数对没有初始化的变量返回 0，但对于空的数组也会返回 0。如果要判断变量是否初始化，则可以应用 isset()函数。另外，count()函数不能识别无限递归。

实例 126 向数组中添加元素

光盘位置：光盘\MR\02\126

高级

趣味指数：★★★★

实例说明

数组元素的个数和元素值是可以改变的。本实例通过数组函数 array_push()向数组中添加元素，运行结果如图 2.100 所示。

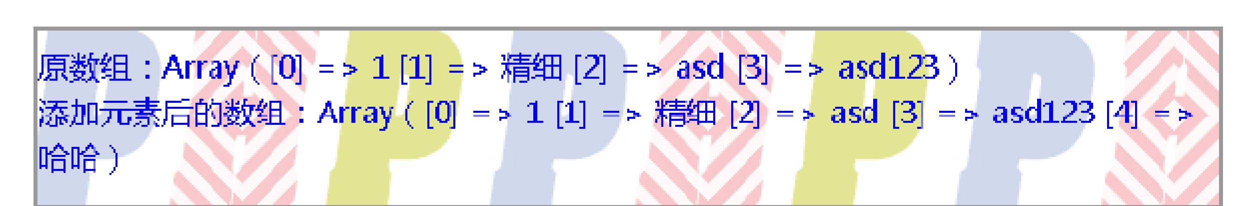
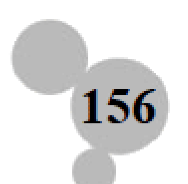


图 2.100 向数组中添加元素



关键技术

在 PHP 中，向数组中添加元素使用 array_push()函数，其语法如下：

```
int array_push ( array array, mixed var [, mixed ...]);
```

参数说明：

array array：原数组。

mixed var：压入的数据元素。

该函数将 array 当成一个栈，并将传入的变量压入 array 的末尾。

设计过程

创建 index.php 文件，首先，定义数组变量，并为数组变量赋值。然后，利用 array_push()函数向数组中添加元素并打印数组输出信息，其代码如下：

```
<?php
	$a = array('1','精细','asd','asd123');				//声明一个数组
	echo "原数组：";
	print_r($a)."<br>";							//输出数组
	array_push($a,'哈哈');						//向数组末尾添加元素
	echo "<br>添加元素后的数组：";
	print_r($a);								//打印添加元素后的数组
?>
```

秘笈心法

心法领悟 126：使用 array_push()函数的相关说明。

如果用 array_push()函数单纯地给数组增加一个单元，不如选用“$array[]=”，因为这样没有调用函数的额外负担。

实例 127	将数组中指定索引位置的元素替换 光盘位置：光盘\MR\02\127	高级 趣味指数：★★★★

实例说明

创建数组时，由于疏忽将元素内容书写错误，更改错误的元素内容可以通过数组键值实现。本实例通过数组下标（键值）实现将数组中指定索引位置的元素替换，运行结果如图 2.101 所示。

原数组元素：Array ([0] => 1 [1] => 精细 [2] => asd [3] => asd123)
指定索引元素替换后的数组：Array ([0] => 1 [1] => 精细 [2] => 经济 [3] => asd123)
键值1的元素发生更改

图 2.101　将数组中指定索引位置的元素替换

关键技术

PHP 数字索引由数字组成，下标从 0 开始，数值索引一般表示数组元素在数组中的位置，数字索引数组默认索引值从数字 0 开始，不需要特别的指定，PHP 会自动为索引数组的键名赋一个整数值，然后从这个值开始自动增量，当然也可以指定某个位置开始保存数据。

设计过程

创建 index.php 文件，定义数组变量并通过数组下标输出元素，其代码如下：



```
<?php
	echo "原数组元素：";
	$a = array('1','精细','asd','asd123');                          //定义数组
	print_r($a);                                                     //打印数组
	echo "<br>指定索引元素替换后的数组：";
	$a[2] = '经济';                                                  //根据数组下标更改元素
	print_r($a);                                                     //打印更改后的数组
	echo "<br>键值 2 的元素发生更改";
?>
```

秘笈心法

心法领悟 127：使用数组键值的注意事项。

当使用“组名称[键值]”这样的格式时，要注意数组键值是从 0 开始的，所以本例中更改键值为 2 的元素，其实是数组中的第 3 个元素。

实例 128 获取数组中最后一个元素

光盘位置：光盘\MR\02\128

高级

趣味指数：★★★★

实例说明

在数组元素比较多的情况下，想要取得数组中的最后一个元素可以通过先计算元素的个数再通过下标取得，还可以通过函数取得。本实例通过数组函数 array_pop()实现获取数组中的最后一个元素，运行结果如图 2.102 所示。



图 2.102　获取数组中最后一个元素

关键技术

使用 array_pop()函数获取并返回 array 数组的最后一个单元，并将数组 array 的长度减 1，其语法如下：

```
mixed array_pop ( array array);
```

设计过程

创建 index.php 文件。首先定义数组变量$a 并为此数组变量赋值，然后利用函数 array_pop()弹出数组中的最后一个元素并输出结果，其代码如下：

```
<?php
	$a = array('a','b','c','d','e','f','g','h','i');                //声明数组
	echo "获取数组中最后一个元素是：".array_pop($a);                 //返回数组中最后一个元素
?>
```

秘笈心法

心法领悟 128：使用 array_pop()函数的相关说明。

在使用 array_pop()函数时，数组中最后一个元素是被取出而不是被复制，即每次使用此函数，数组元素就会减少一个。如果数组为空（或者不是数组）将返回 NULL。

实例 129 去除数组中的重复元素

光盘位置：光盘\MR\02\129

高级

趣味指数：★★★★

实例说明

在数组中键值是唯一的，但是元素的值是可以重复的。想要删除数组元素中重复的值可以使用函数



array_unique()。本实例是通过数组函数 array_unique()去除数组中的重复元素值，运行结果如图 2.103 所示。

原数组元素为：Array ([0] => a [1] => b [2] => a [3] => d [4] => a [5] => f [6] => a [7] => h [8] => a)
去除重复项后的数组为：Array ([0] => a [1] => b [3] => d [5] => f [7] => h)

图 2.103　去除数组中的重复元素

关键技术

在 PHP 中，通过 array_unique()函数去除数组中的重复元素，其语法如下：

```
array array_unique ( array array);
```

参数 array 为指定参数的数组，其返回值为一个没有重复元素的新数组。

设计过程

创建 index.php 文件。首先创建数组变量$a 并为其赋值，然后利用 array_unique()函数去除数组中的重复元素值，最后打印数组，其代码如下：

```
<?php
	$a = array('a','b','a','d','a','f','a','h','a');		//定义数组
	echo "原数组元素为：";
	print_r($a);							//打印数组
	echo "<br>去除重复项后的数组为：";
	print_r(array_unique($a));				//输出去除重复项的数组
?>
```

秘笈心法

心法领悟 129：使用 array_unique()函数的原理。

使用 array_unique()函数的原理是先将值作为字符串排序，然后对每个值只保留第一个遇到的键名，而忽略所有后面的键名。但这并不意味着在未排序的数组中同一个值的第一个出现的键名会被保留。

实例 130　字符串与数组的转换

光盘位置：光盘\MR\02\130

高级

趣味指数：★★★★

实例说明

实例 109 介绍了字符串与数字之间的转换，下面来介绍字符串与数组之间的转换。本实例通过强制类型转换实现字符串与数组的转换，运行结果如图 2.104 所示。



图 2.104　字符串与数组的转换

关键技术

虽然 PHP 是弱类型语言，但有时仍然需要用到类型转换。PHP 中的类型转换和 C 语言一样，非常简单，在变量前加上用括号括起来的类型名称即可。允许转换的类型如表 2.7 所示。



表 2.7　类型强制转换

转换操作符	转 换 类 型	举　例
(boolean)	转换成布尔型	(boolean)$num、(boolean)$str
(string)	转换成字符型	(string)$boo、(string)$flo
(integer)	转换成整型	(integer)$boo、(integer)$str
(float)	转换成浮点型	(float)$str、(float)$str
(array)	转换成数组	(array)$str
(object)	转换成对象	(object)$str

设计过程

创建 index.php 文件。首先定义数组变量并为此变量赋值，然后利用强制类型转换的方法将字符串类型转换为数组类型，其代码如下：

```
<?php
	$a = "123";					//声明字符串变量
	$b = (array)$a;				//将字符串转换为数组
	echo "字符串转换为数组：";
	print_r($b);					//打印数组
	echo "<hr>";
	$c = array('1');				//定义数组
	$d = (string)$c;				//将数组转换为字符串
	echo"<br>数组转换为字符串：";
	echo $d;						//输出字符串
?>
```

秘笈心法

心法领悟 130：将数组强制转换成字符串的注意事项。

将数组强制转换成字符串时，数组将被转换成字符串 "Array"，因此无法通过 echo()或者 print()函数来输出数组的内容。

实例 131　对数组元素进行随机排序

光盘位置：光盘\MR\02\131

高级

趣味指数：★★★★

实例说明

本实例通过 shuffle()函数实现对数组元素进行随机排序，运行结果如图 2.105 所示。

原数组元素顺序为：经济 a b 123 asd
经随机排序后：a 经济 asd 123 b

图 2.105　对数组元素进行随机排序

关键技术

在 PHP 中，将数组元素进行随机排序使用 shuffle()函数，其语法如下：

```
void shuffle ( array array);
```

使用该函数将数组打乱，随机排序。

设计过程

新建 index.php 文件。首先定义数组变量，利用 srand()函数播下随机数发生器种子，然后利用 shuffle()函数



将数组打乱，最后输出数组元素，其代码如下：

```
<?php
$numbers = array('经济','a','b','123','asd');
echo "原数组元素顺序为：经济  a b 123 asd";
srand ((float)microtime()*1000000);
shuffle ($numbers);
echo "<br>经随机排序后：";
while (list (, $number) = each ($numbers)) {
		echo "$number ";
}
?>
```

秘笈心法

心法领悟 131：使用 shuffle()函数的注意事项。

在使用 shuffle()函数将数组打乱时，必须用 srand()函数播下本函数的随机数发生器的种子，否则得不到预期结果。

实例 132　随机抽取数组中的元素　光盘位置：光盘\MR\02\132　高级　趣味指数：★★★★

实例说明

可以把随机抽取数组中的元素看成一个彩票抽奖的大摇箱，在原理上两者是没什么区别的，但是彩票抽奖的大摇箱可能存在一点点的偶然性，例如大摇箱里的球弹力不同等。本实例通过随机函数 rand()实现随机抽取数组中的元素，运行结果如图 2.106 所示。

数组：Array ([0] => 经济 [1] => a [2] => b [3] => 123 [4] => asd)
随机取得数组的元素是：b

图 2.106　随机抽取数组中元素

关键技术

应用 rand()函数获取随机数值，并将随机数值指向数组下标从而实现随机获取指定元素的值。rand()函数的语法如下：

```
int rand ( [int min, int max]);
```

参数说明：

min：随机数的最小值。

max：随机数的最大值。

设计过程

创建 PHP 脚本。首先定义数组变量并为数组变量赋值，然后利用 rand()函数获取 0~4 中的一个随机数，并将此随机数赋给数组变量作为数组变量的键值，最后输出此数组变量，其代码如下：

```
<?php
	$numbers = array('经济','a','b','123','asd');			//定义数组
	echo "数组：";
	print_r($numbers);
	$rand = rand(0,4);						//创建一个 0~4 的随机数字
	echo "<br>随机取得数组的元素是：".$numbers[$rand];	//将随机数字作为键值获取数组元素
?>
```

秘笈心法

心法领悟 132：随机函数 rand()的相关说明。

在使用随机函数 rand()时，如果没有提供可选参数 min 和 max，该函数返回 0~RAND_MAX 之间的伪随机数。例如，想得到 10~30（包括 10 和 30）之间的随机数，使用 rand(5, 15)。

实例 133　二维数组的输出

光盘位置：光盘\MR\02\133

高级

趣味指数：★★★★

实例说明

二维数组可以简单理解为数组元素中还包含一个数组。MySQL 图形化管理工具 phpMyAdmin 中存储的每个数据表其实就是一个二维数组。本实例实现二维数组中数据的输出，运行结果如图 2.107 所示。

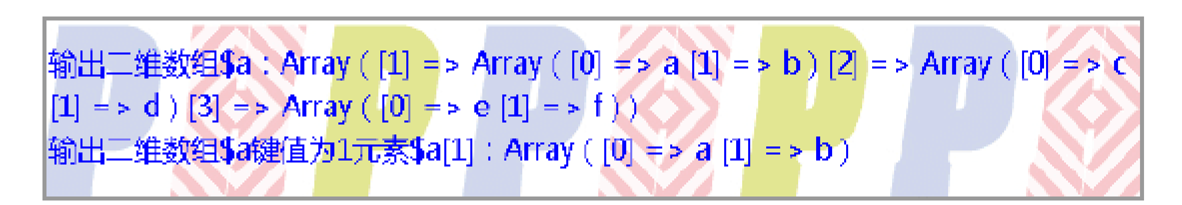


图 2.107　二维数组的输出

关键技术

如果创建的数组中的元素值都是变量，此数组称为一维数组；如果创建的数组元素值仍然是数组，那么就称为二维数组。也就是说，一个数组的元素如果仍是一个一维数组，则称这个数组是二维数组。

设计过程

创建 index.php 文件。首先定义二维数组变量$a 并为此变量赋值，然后利用 print_r()函数实现二维数组的输出，其代码如下：

```
<?php
	echo '输出二维数组$a：';
	$a = array(1=>array('a','b'),2=>array('c','d'),3=>array('e','f'));		//创建一个二维数组
	print_r($a)."<br>";
	echo '<br>输出二维数组$a 键值为 1 元素$a[1]：';
	print_r($a[1]);								//打印二维数组键值为 1 的元素
?>
```

秘笈心法

心法领悟 133：对 PHP 中数组的理解。

PHP 中的数组实际上是一个有序图。该图是一种把 values 映射到 keys 的类型，此类型在很多方面做了优化，因此可以把它当成真正的数组来使用，或列表（矢量）、散列表（是图的一种实现）、字典、集合、栈、队列以及更多可能性。因为可以用另一个 PHP 数组作为值，也可以很容易地模拟树。

实例 134　获取数组当前的键名和值

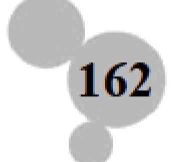
光盘位置：光盘\MR\02\134

高级

趣味指数：★★★★

实例说明

每个数组元素都是由键名和值两部分组成的，想要获取它们可以通过循环来实现。本实例通过 foreach 语句



获取数组当前的键名和值，运行结果如图 2.108 所示。

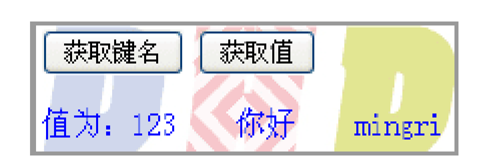


图 2.108　获取键名值

关键技术

数组中的每个实例都包含两项：键和值。其中键可以是数字、字符串或者数字和字符串的组合，用于标识数组中相应的值；而值被称为数组中的元素，可以定义为任意数据类型，甚至是混合类型，最终通过键来获取相应的值。

设计过程

新建 index.php 文件。首先，定义数组变量$array 并且为其赋值。当单击“获取键名”按钮时，利用 foreach 语句遍历数组并输出所有元素的键值；当单击“获取值”按钮时，同样通过 foreach 语句获取数组元素的 value 值并且输出，其代码如下：

```
<?php
	$array =array("编程"=>"123","soft"=>"你好","456"=>"mingri");			//定义数组
	if($_POST[sub]){										//通过 POST 方式传递参数
		echo "键名为：";
		foreach($array as $key => $value){						//遍历数组
			echo $key."    ";				//输出键值
		}
	}
	if($_POST[sub1]){										//通过 POST 方式传递参数
		echo "值为：";
		foreach($array as $key => $value){						//遍历数组
			echo $value."    ";				//输出值
		}
	}
?>
```

秘笈心法

心法领悟 134：使用 foreach 语句的注意事项。

foreach 语句只能用于数组。

实例 135	检测数组中是否存在某个值 光盘位置：光盘\MR\02\135	高级 趣味指数：★★★★

实例说明

PHP 大型的程序开发往往使程序员焦头烂额，有时程序员不确定自己定义的数组中是否存在需要的元素。本实例通过 array_key_exists()函数检测数组中是否存在某个值，运行结果如图 2.109 所示。

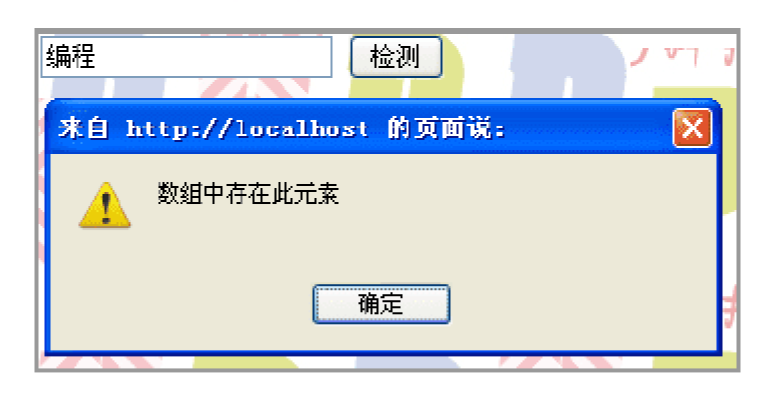


图 2.109　检测数组中是否存在某元素



关键技术

在 PHP 中，检测数组中是否存在某个值，使用 array_key_exists()函数，其语法如下：

```
bool array_key_exists ( mixed key, array search );
```

参数说明：

mixed key：检测元素。

array search：源数组。

该函数检查给定的键名或索引是否存在于数组中。

设计过程

新建 index.php 文件，定义数组变量$array 并为其赋值，当单击“检测”按钮时，利用 array_key_exists()函数判断接收到的文本框信息是否存在于指定的数组元素中，并且根据返回值给出相应的提示信息，其代码如下：

```
<?php
        $array =array("编程"=>"123","soft"=>"你好","456"=>"mingri");            //定义数组
        if($_POST[sub]){                                                          //通过 POST 方式传递参数
                if(array_key_exists($_POST[text],$array)){                        //检测是否存在元素
                        echo "<script>alert('数组中存在此元素');</script>";          //提示
                }else{
                        echo "<script>alert('数组中不存在此元素');</script>";        //提示
                }
        }
?>
```

秘笈心法

心法领悟 135：array_key_exists()函数的相关说明。

array_key_exists()函数在给定的 key 存在于数组中时返回 TRUE，key 可以是任何能作为数组索引的值。array_key_exists()函数也可用于对象。

实例 136	获取数组中的当前单元 光盘位置：光盘\MR\02\136	高级 趣味指数：★★★★

实例说明

本实例通过 current()函数获取数组中的当前单元，运行结果如图 2.110 所示。

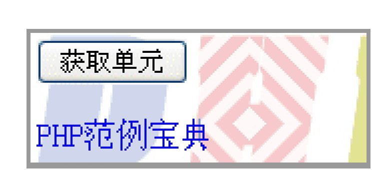


图 2.110　获取数组中的当前单元

关键技术

current()函数的主要功能是返回数组中的当前单元，其语法如下：

```
mixed current ( array &array );
```

设计过程

（1）创建 PHP 脚本文件。首先，定义数组变量并为其赋值，当单击“获取单元”按钮时，通过 current()函数获取数组当前元素，其代码如下：

```
<?php
        $array =array("编程"=>"PHP 范例宝典","soft"=>"你好","456"=>"mingri");            //定义数组
```



```
    if($_POST[sub]){                                       //通过 POST 方式传递参数
        echo current($array);                              //获取当前数组中的单元
    }
?>
```

（2）将该文件存储于\MR\02\136 文件夹中，命名为 index.php。运行结果如图 2.110 所示。

秘笈心法

心法领悟 136：使用 current()函数的相关说明。

current()函数返回当前被内部指针指向的数组单元的值，并不移动指针。如果内部指针指向超出了单元列表的末端，current()函数返回 FALSE。如果数组包含有空的单元（0 或者""，空字符串），则本函数在碰到这个单元时也返回 FALSE。

实例 137　从数组中随机取出元素

光盘位置：光盘\MR\02\137

高级　趣味指数：★★★★

实例说明

通过 rand()函数可以随机抽取数组中的元素。下面再通过另外一个函数 array_rand()实现从数组中随机取出元素，运行结果如图 2.111 所示。

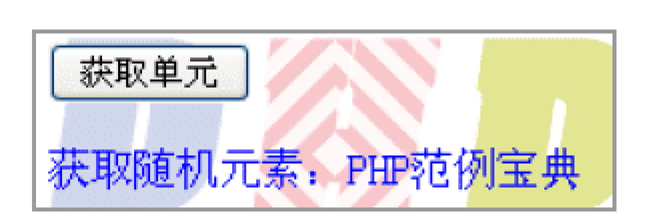


图 2.111　从数组中随机取出元素

关键技术

当用户想从数组中取出一个或多个单元时，使用 array_rand()函数是非常有用的，它与 rand()函数获取随机数组元素的原理大同小异，其语法如下：

```
mixed array_rand ( array input [, int num_req] );
```

参数说明：

array input：传入的数组。

int num_req：可选参数，定义输出数组元素的个数。

设计过程

（1）创建 PHP 脚本文件。首先，定义数组变量$array 并为其赋值，当单击“获取单元”按钮时，通过 array_rand()函数获取数组中的随机元素值，其代码如下：

```
<?php
    $array =array("编程"=>"PHP 范例宝典","soft"=>"明日科技","456"=>"mingri");    //定义数组
    if($_POST[sub]){                                                        //通过 POST 方式传递参数
        $rand_keys = array_rand($array, 1);                                 //获取数组的随机键值
        echo "获取随机元素：".$array[$rand_keys] ;                            //输出随机元素值
    }
?>
```

（2）将该文件存储于\MR\02\137 文件夹中，命名为 index.php。运行结果如图 2.111 所示。

秘笈心法

心法领悟 137：使用 array_rand()函数的相关说明。

array_rand()函数在从数组中取出一个或多个随机的元素时相当有用。它接受 $array 作为输入数组和一个可选的参数 num_req，指明想取出多少个元素。如果没有指定，默认为 1。如果只取出一个，array_rand()函数返回一个随机元素的键名，否则就返回一个包含随机键名的数组。这样用户就可以随机从数组中取出键名和值。

实例 138　合并数组

光盘位置：光盘\MR\02\138

高级

趣味指数：★★★★

实例说明

数组与数组之间是彼此独立的，但是两个数组可以合并到一起成为一个新数组。本实例通过数组函数 array_merge()实现合并数组，运行结果如图 2.112 所示。

合并数组

数组1：Array ([编程] => PHP范例宝典 [soft] => 明日科技 [456] => mingri)

数组2：Array ([0] => PHP编程 [1] => Java编程 [2] => C#编程 [3] => ASP.NET编程)

合并后的数组Array ([编程] => PHP范例宝典 [soft] => 明日科技 [0] => mingri [1] => PHP编程 [2] => Java编程 [3] => C#编程 [4] => ASP.NET编程)

图 2.112　合并数组

关键技术

数组的合并应用的是 array_merge()函数，其语法如下：

```
array array_merge ( array array1 [, array array2 [, array ...]] );
```

参数说明：

array array1：数组 1。

array array2：数组 2。

设计过程

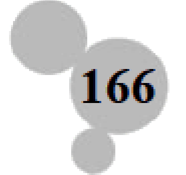
新建 index.php 文件。首先，定义两个数组变量并为其赋值。当单击“合并数组”按钮时，利用 array_merge()函数合并数组，并打印合并后的数组，其代码如下：

```
<?php
	$array =array("编程"=>"PHP 范例宝典","soft"=>"明日科技","456"=>"mingri");	//定义数组 1
	echo "数组 1：";
	print_r($array);	//打印数组
	echo "<br><br>";
	$arr = array("PHP 编程","Java 编程","C#编程","ASP.NET 编程");	//定义数组 2
	echo"数组 2：";
	print_r($arr);	//打印数组
	echo "<br><br>";
	if($_POST[sub]){	//通过 POST 方式传递参数
		echo "合并后的数组";
		print_r(array_merge($array,$arr));	//打印合并后的数组
	}
?>
```

秘笈心法

心法领悟 138：使用 array_merge()函数的相关说明。

array_merge()函数将一个或多个数组的单元合并起来，一个数组中的值附加在前一个数组的后面，返回作为结果的数组。如果输入的数组中有相同的字符串键名，则该键名后面的值将覆盖前一个值。如果数组包含数



字键名，后面的值将不会覆盖原来的值，而是附加到后面。如果只给了一个数组并且该数组是数字索引的，则键名会以连续方式重新索引。

实例 139　**拆分数组**　　高级

光盘位置：光盘\MR\02\139　　趣味指数：★★★★

实例说明

存在数组的合并函数当然也就存在数组的拆分函数。本实例通过 array_chunk()函数实现数组的拆分，运行结果如图 2.113 所示。

拆分数组

原数组：Array ([0] => PHP编程 [1] => Java编程 [2] => C#编程 [3] => ASP.NET编程)

拆分后的数组：Array ([0] => Array ([0] => PHP编程 [1] => Java编程) [1] => Array ([0] => C#编程 [1] => ASP.NET编程))

图 2.113　拆分数组

关键技术

array_chunk()函数可将一个数组分割成多个，其原理是将原有数据根据传递的参数不同分割成多个二维数组。array_chunk()函数的语法如下：

```
array array_chunk ( array input, int size [, bool preserve_keys] );
```

参数说明：

array input：源数组。

int size：分割成指定数量的数组。

设计过程

（1）创建 PHP 脚本文件。首先，定义数组变量并为其赋值，当单击“拆分数组”按钮时，打印被拆分后的数组，其代码如下：

```
<?php
	$arr = array("PHP 编程","Java 编程","C#编程","ASP.NET 编程");			//定义数组
	echo"原数组：";
	print_r($arr);										//打印数组
	echo "<br><br>";
	echo "拆分后的数组：";
	if($_POST[sub]){									//通过 POST 方式传递参数
		print_r(array_chunk($arr,2))."<br>";					//拆分数组
	}
?>
```

（2）将该文件存储于\MR\02\139 文件夹中，命名为 index.php。运行结果如图 2.113 所示。

秘笈心法

心法领悟 139：使用 array_chunk()函数的注意事项。

使用 array_chunk()函数拆分的数组，最后一个数组的单元数目可能会少几个。

实例 140

遍历数组

光盘位置：光盘\MR\02\140

高级

趣味指数：★★★★

实例说明

通过 for 语句可以遍历数组，下面通过另外一种方法来遍历数组。本实例通过 list()和 each()函数实现遍历数组，运行结果如图 2.114 所示。

遍历数组

Array ([1] => PHP编程 [value] => PHP编程 [0] => 0 [key] => 0)
Array ([1] => Java编程 [value] => Java编程 [0] => 1 [key] => 1)
Array ([1] => C#编程 [value] => C#编程 [0] => 2 [key] => 2)
Array ([1] => ASP.NET编程 [value] => ASP.NET编程 [0] => 3 [key] => 3)

图 2.114　遍历数组

关键技术

遍历数组的方法有很多，除了使用 foreach 语句外，还可以通过 list()和 each()函数配合使用来实现，其相关语法如下：

list()函数把数组中的值赋给一些变量。与 array()函数类似，其不是真正的函数而是语言结构。list()函数仅能用于数字索引的数组，且数字索引从 0 开始，其语法如下：

```
void list ( mixed ...)
```

参数 mixed 为被赋值的变量名称。

each()函数，返回数组中当前指针位置的键名和对应的值，并向前移动数组指针。键值对被返回为 4 个单元的数组，键名为 0、1、key 和 value。单元 0 和 key 包含有数组单元的键名，1 和 value 包含有数据，其语法如下：

```
array each ( array array)
```

设计过程

（1）创建 PHP 脚本文件，当单击“遍历数组”按钮时，首先定义数组变量并为其赋值，然后通过 list()和 each()函数实现数组遍历并输出结果，其代码如下：

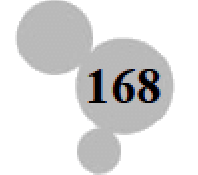
```
<?php
	if($_POST[sub]){                                                    //通过 POST 方式传递参数
		$foo = array("PHP 编程","JAVA 编程","C#编程","ASP.NET 编程");     //定义数组
		for($a = 0;$a < count($foo);$a++){                              //定义循环
			print_r(list($name, $value)=each($foo));                    //打印数组
		}
}
?>
```

（2）将该文件存储于\MR\02\140 文件夹中，命名为 index.php。运行结果如图 2.114 所示。

秘笈心法

心法领悟 140：使用 list()和 each()函数的注意事项。

list()函数只能用于数字索引的数组。如果内部指针越过了数组的末端，则 each()函数返回 FALSE。



2.11　日期和时间

对日期和时间进行操作的程序无处不在，特别是在基于 Web 的应用程序中，如表单提交的时间、用户登录的时间、数据库中数据的更新和删除的时间等。想要记录这些操作执行的时刻，就需要通过日期和时间来完成。日期和时间在人们的生活中必不可少，同样，在互联网中也是非常重要的。

实例 141	设置系统的当前时间 光盘位置：光盘\MR\02\141	高级 趣味指数：★★★★

实例说明

系统的当前时间受时区限制，默认情况下是格林威治时间。用户正确取得本地时间是很重要的。本实例通过 date()函数获取系统的当前时间，运行结果如图 2.115 所示。



图 2.115　设置系统的当前时间

关键技术

因为在 PHP 语言中，日期、时间函数依赖于服务器的地区设置，而 PHP 默认设置的是标准的格林威治时间（即采用的是零时区），所以，如果没有对 PHP 的时区进行设置，那么使用日期、时间函数获取的将是英国伦敦本地时间（即零时区的时间）。

这也是对 PHP 的时区进行设置的原因，因为如果不设置正确的时区，那么 PHP 的日期、时间函数就获取不到正确的当地时间。例如，以东八区为例，如果当地使用的是北京时间，那么在没有对 PHP 的时区进行设置时，获取的时间就将比北京时间少 8 个小时。

更改 PHP 语言中的时区设置有如下两种方法。

（1）在 php.ini 文件中，定位到[date]下的“;date.timezone =”选项，去掉前面的分号，并设置它的值为当地所在时区使用的时间。修改内容如图 2.116 所示。

例如，如果当地所在时区为东八区，那么就可以设置“date.timezone =”的值为 PRC、Asia/Hong_Kong、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐）等，这些都是东八区的时间。



图 2.116　设置 PHP 的时区

设置完成后，保存文件，重新启动 Apache 服务器。

（2）在应用程序中，在日期、时间函数之前使用 date_default_timezone_set()函数就可以完成对时区的设置。date_default_timezone_set()函数的语法如下：

```
date_default_timezone_set(timezone);
```

参数 timezone 为 PHP 可识别的时区名称，如果无法识别，则系统采用 UTC 时区。



例如，设置北京时间可以使用的时区包括 PRC（中华人民共和国）、Asia/Chongqing（重庆）、Asia/Shanghai（上海）和 Asia/Urumqi（乌鲁木齐），这几个时区名称是等效的。

设计过程

新建 index.php 文件，当单击“地区时间”按钮时，首先为当前 PHP 服务时间设置时区，并通过 switch 语句根据地址栏传递参数的不同，通过 date()函数获取不同国家和地区的时间，其代码如下：

```
<?php
	if($_POST[sub]){                                      //通过 POST 方式传递参数
		date_default_timezone_set("Asia/ShangHai");       //设置时区
		switch($_POST[select]){                           //条件语句
			case "中国";                                  //如果是中国
			echo date("H:i:s");                           //输出时间
			break;                                        //结束语句
			case "英国";                                  //如果是英国
			echo (date("H")-8).":".date("i:s");           //输出时间
			break;                                        //结束语句
			case "日本";                                  //如果是日本
			echo(date("H")+1).":".date("i:s");            //输出时间
			break;                                        //结束语句
			case "开罗";                                  //如果是开罗
			echo(date("H")-6).":".date("i:s");            //输出时间
			break;                                        //结束语句
			case "莫斯科";                                //如果是莫斯科
			echo(date("H")-5).":".date("i:s");            //输出时间
			break;                                        //结束语句
		}
	}
?>
```

秘笈心法

心法领悟 141：知识扩展。

本实例只是简单地向用户介绍时区及不同地区相差的小时数，在实际开发中不能对时间单纯地进行算术运算，需要使用关键技术中提供的方法对 PHP 的时区进行设置。

实例 142 将日期和时间转换为时间戳

光盘位置：光盘\MR\02\142

高级

趣味指数：★★★★

实例说明

将日期转换为时间戳是将时间进行数学运算的重要手段。本实例通过 time()函数实现将日期和时间转换为时间戳，运行结果如图 2.117 所示。

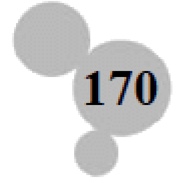


图 2.117　将日期和时间转换为时间戳

关键技术

获取系统 UNIX 时间戳的方法有很多，使用 time()函数是比较常用的一种方法，time()函数的语法如下：

```
int time ( void );
```



该函数没有参数，返回值为 UNIX 时间戳的整数值。

设计过程

创建 index.php 文件。首先，设置 PHP 的时区为中国上海，然后，利用 time()函数获取当前的时间戳，并通过 echo 语句输出时间戳，其代码如下：

```
<?php
date_default_timezone_set("Asia/ShangHai");                    //设置时区
echo "当前时间：".date('Y-m-d H:i:s');                          //取得当前时间
echo "<br>转换成时间戳为：".time();                              //转换为时间戳
?>
```

秘笈心法

心法领悟 142：知识扩展。

将当前日期和时间转换为时间戳的方法有很多，下面通过 strtotime()函数实现本例功能，代码如下：

```
<?php
date_default_timezone_set("Asia/ShangHai");                    //设置时区
echo "当前时间：".date('Y-m-d H:i:s');                          //设置当前日期和时间
echo "转换时间戳为：”strtotime("now");                          //转换时间戳
?>
```

实例 143　获取系统中的特定日期和时间

光盘位置：光盘\MR\02\143

高级

趣味指数：★★★★

实例说明

在 PHP 程序设计中，日期和时间是具有一定格式的，例如，今天是一年的第多少天等。本实例应用 date()函数通过传入参数不同实现日期和时间的格式化，运行结果如图 2.118 所示。

当前时间：2015年01月20日 15点48分37秒 PM
本月有31天
今天是一周的第2天,是一年的第19天

图 2.118　日期和时间的格式化

关键技术

本实例通过 date()函数实现日期和时间的格式化，详细说明请参见实例 035。

设计过程

（1）创建 PHP 脚本文件，设置 PHP 的时区为中国上海。然后，利用向 date()函数中传递的参数不同实现对日期和时间的格式化，并输出格式化后的信息，其代码如下：

```
<?php
date_default_timezone_set("Asia/ShangHai");                    //设置时区
echo "当前时间：".date('Y 年 m 月 d 日 H 点 i 分 s 秒 A')."<br>";    //格式化日期和时间
echo "本月有".date('t')."天.<br>今天是一周的第".date('w')."天,是一年的第".date('z')."天";
?>
```

（2）将该文件存储于\MR\02\143 文件夹下，命名为 index.php。运行结果如图 2.118 所示。

秘笈心法

心法领悟 143：知识扩展。

date()函数还有很多其他的参数，熟悉这些参数的使用可以在使用格式化时间、日期上事半功倍，具体参数

说明请参见实例 035。

实例 144　比较时间的大小

光盘位置：光盘\MR\02\144

高级

趣味指数：★★★★

实例说明

在 PHP 语言中，要完成对两个时间大小的比较，必须先将时间转换为时间戳，可以通过 strtotime()函数完成。本实例应用 strtotime()函数对两个固定的时间进行比较，其运行结果如图 2.119 所示。



图 2.119　比较时间的大小

关键技术

PHP 中应用 strtotime()函数将任何英文文本的日期解析为 UNIX 时间戳，其值为相对于 now 参数给出的时间，如果没有提供此参数则用系统当前时间，此时与实例 142 中的 time()函数相同。strtotime()函数的语法如下：

```
int strtotime ( string time [, int now] );
```

该函数有两个参数。如果参数 time 的格式是绝对时间，则 now 参数不起作用；如果参数 time 的格式是相对时间，其对应的时间就是参数 now 来提供的，当没有提供参数 now 时，对应的时间就为当前时间。如果解析失败，则返回 FALSE。在 PHP 5.1.0 之前的版本中，本函数在失败时返回-1。

设计过程

（1）创建 PHP 脚本文件。首先，编写 form 表单并设置当前时区为中国上海，当单击“日期比较”按钮时，运用 strtotime()函数获取两个不同时间点的时间戳，并进行求差运算。最后，输出相关提示，其代码如下：

```
<?php
	echo "<form action='' method='post'>";                    //输出表单
	echo "<input type='submit' name='sub' value='日期比较'>";
	echo "</form>";
	date_default_timezone_set("Asia/ShangHai");              //设置时区
	if($_POST[sub]){                                          //通过 POST 方式传递参数
		$time = date("Y 年 m 月 d 日");                       //取得当前日期
		$time_1 = time();                                     //取得当前时间戳
		$time_2 = strtotime("27 June 2014");                  //取得固定日期的时间戳
		if($time_2 < $time_1){                                //时间戳比较
			echo $time."比 2014 年 6 月 27 日大";
		}else{
			echo $time."比 2014 年 6 月 27 日小";
		}
	}
?>
```

（2）将该文件存储于\MR\02\144 文件夹下，命名为 index.php。运行结果如图 2.119 所示。

秘笈心法

心法领悟 144：使用 strtotime()函数的相关说明。

strtotime()函数预期接受一个包含美国英语日期格式的字符串，并尝试将其解析为 UNIX 时间戳（自 January 1 1970 00:00:00 GMT 起的秒数），其值相对于 now 参数给出的时间，如果没有提供此参数则用系统当前时间。



实例 145　计算考试时间

光盘位置：光盘\MR\02\145

高级

趣味指数：★★★★

实例说明

计算考试时间程序是单纯地将两个时间点的时间戳做算术运算。本实例通过time()函数实现考试时间的计算，运行结果如图 2.120 所示。

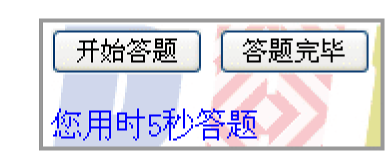


图 2.120　计算考试时间

关键技术

本实例的关键点是 time()函数的灵活运用。当单击“开始答题”按钮时，将 time()函数获取的时间戳存储到 SESSION 变量中；当单击“答题完毕”按钮时，再次通过 time()函数获取系统的当前时间戳。当前时间戳减去开始定义的时间戳，就是考试所用时间。

设计过程

（1）创建 PHP 脚本文件。首先编写 form 表单并设置两个提交按钮，当每个按钮被单击后，利用 SESSION 变量保存此时的时间戳并进行求差运算，然后输出结果，其代码如下：

```
<?php
	echo "<form action ='' method ='post'>";										//输出表单
	echo "<input type = 'submit' name='sub' value='开始答题'> <input type='submit' name= 'sub_1'value ='答题完毕'>";
	echo "</form>";
	if($_POST[sub]){															//通过 POST 方式传递参数
		$time = time();															//取得当前时间戳
		$_SESSION['time'] = $time;												//将时间戳保存在 SESSION 中
		echo "<script>alert('单击确定开始答题');</script>";						//输出提示
	}
	if($_POST[sub_1]){															//通过 POST 方式传递参数
		if(isset($_SESSION['time'])){											//判断 SESSION 是否存在
			echo "您用时".(time()-$_SESSION['time'])."秒答题";					//输出结果
			session_destroy();													//销毁 SESSION
		}else{																	//否则输出提示
			echo "<script>alert('您还没有答题无法结束，请先答题！');location.href='index.php'</script>";
		}
	}
?>
```

（2）将该文件存储于\MR\02\145 文件夹下，命名为 index.php。运行结果如图 2.120 所示。

秘笈心法

心法领悟 145：知识扩展。

本实例运用 SESSION 机制保存考试的开始时间，然后完成考试时间的计算。有关 SESSION 机制的详细讲解请参考本书第 5 章的内容。

实例 146　倒计时

光盘位置：光盘\MR\02\146

高级

趣味指数：★★★★

实例说明

倒计时功能的实现同样也是将时间戳做算术运算。本实例应用 strtotime()函数实现倒计时程序，运行结果如图 2.121 所示。

当前时间：2015年01月20日 16点00分06秒
距2016年元旦还有346天

图 2.121　倒计时

关键技术

本实例的关键点是 strtotime()函数的灵活运用，详细说明参见实例 144。

设计过程

（1）创建 PHP 脚本文件，设置当前时区为中国上海。首先，利用 time()函数取得当前时刻的时间戳并保存在变量中。然后，利用 strtotime()函数取得 2011 年 1 月 1 日的时间戳并保存在变量中，将两个时间戳进行求差运算，将结果转换成天数后输出，其代码如下：

```
<?php
    date_default_timezone_set("Asia/ShangHai");                                    //设置时区
    echo "当前时间：".date('Y 年 m 月 d 日  H 点 i 分 s 秒')."<br>";                //输出当前日期
    $time = time();                                                               //取得当前时间戳
    $time_r = strtotime("1 January 2016");                                        //取得指定日期的时间戳
    echo "距 2016 年元旦还有<b style='color:red;'>".ceil((($time_r - $time)/(3600*24)))."</b>天";  //输出
?>
```

（2）将该文件存储于\MR\02\146 文件夹下，命名为 index.php。运行结果如图 2.121 所示。

秘笈心法

心法领悟 146：时间比较的原理。

通过上面几个实例用户应该明白，其实所有的时间比较归根结底都是时间戳的比较，所以掌握时间戳是十分重要的。

实例 147　网页闹钟

光盘位置：光盘\MR\02\147

高级　趣味指数：★★★★

实例说明

闹钟是人们生活中经常使用的一个小工具，它会在指定时间叫你起床。在 Web 程序中，也可以应用这个原理开发一个“网页闹钟”，提示用户在指定的时间或者时间段内要做什么工作。本实例通过 mktime()函数实现网页闹钟程序，运行结果如图 2.122 所示。

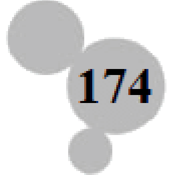


图 2.122　网页闹钟

关键技术

PHP 中，应用 mktime()函数将一个时间转换成 UNIX 的时间戳值。mktime()函数根据给出的参数返回 UNIX 时间戳。时间戳是一个长整数，包含从 UNIX 纪元到给定时间的秒数。mktime()函数的语法如下：

```
int mktime ( [int hour [, int minute [, int second [, int month [, int day [, int year [, int is_dst]]]]]]] );
```



mktime()函数的参数说明如表 2.8 所示。

表 2.8　mktime()函数的参数说明

参　数	说　明
hour	小时数
minute	分钟数
second	秒数（一分钟之内）
month	月份数
day	天数
year	年份数，可以是 2 位或 4 位数字，0~69 对应于 2000~2069，70~100 对应于 1970~2000
is_dst	在夏令时可以被设为 1，如果不是则设为 0；如果不确定是否为夏令时，则设为-1（默认值）

设计过程

（1）创建 index.php 文件。首先定义 PHP 时区为中国上海，然后编写 form 表单，通过 mktime()函数实现获取当前的时间戳，其代码如下：

```
<?php
	echo "<table border='1' bordercolor='#FF0000' cellspacing='0' align='center'><tr><td>";	//输出表格
	echo "<h2 style='color:#0033FF'>闹钟程序</h2>";	//输出标题
	date_default_timezone_set("Asia/ShangHai");	//设置时区
	echo "<h6 style='color:red'>今天是".date("Y-m-d H:i:s")."</h6>";	//输出当前时间
	echo "<form action='' method='post'>";	//输出表单
	echo "<input type='text' name='text1' size='2'>年<input type='text' name='text2' size='2'>月";
	echo "<input type='text' name='text3' size='2'>日<br><input type='text' name='text4' size='2'>时";
	echo "<input type='text' name='text5' size='2'>分<input type='text' name='text6' size='2'>秒<br>";
	echo "<input type='submit' name='sub' value='定时'>";
	echo "</form>";
	$a = $_POST[text1];	//取得文本框内容
	$b = $_POST[text2];	//取得文本框内容
	$c = $_POST[text3];	//取得文本框内容
	$d = $_POST[text4];	//取得文本框内容
	$e = $_POST[text5];	//取得文本框内容
	$f = $_POST[text6];	//取得文本框内容
	$time = date(mktime($d, $e, $f, $b, $c, $a));	//获取时间戳
	$_SESSION['time'] = $time;	//保存在 SESSION 中
	if($_POST[sub]){	//通过 POST 方式传递参数
		echo"<script>window.location.href='in.php';</script>";	//跳转
	}
	echo "</td></tr></table>";
?>
```

（2）创建 in.php 文件，其代码如下：

```
<?php
	date_default_timezone_set("Asia/ShangHai");	//设置时区
	$time_1 = time();	//取得当前时间戳
	if($_SESSION['time'] < $time_1){	//判断条件
		echo "<script>alert('时间一去不复返');location.href='index.php'</script>";	//输出提示
	}else{
		if($_SESSION['time'] == $time_1){	//判断条件
			echo "<script>alert('今天是 XXX 生日');</script>";	//输出提示
		}else{	//输出提示
			echo "距离 XXX 的生日还有".ceil((($_SESSION['time']-$time_1)/(3600*24)))."天";
		}
	}
?>
```

秘笈心法

心法领悟 147：不同版本的操作平台对 UNIX 时间戳的支持。

PHP 为 UNIX 时间戳的处理提供各种函数。到目前的 PHP 版本为止，由于任何已知 Windows 版本以及一些其他系统均不支持负的时间戳，因此在 Windows 中无法表示 1970 年 1 月 1 日之前的时间。因为目前 UNIX 时间戳是以 32 位二进制表示的，其范围为-2147483648～+2147483647），因此，目前 UNIX 时间戳可表示的最大时间为 2038 年 1 月 19 日 3 点 14 分 7 秒，该时刻时间戳为 2147483647，对于该时刻之后的时间，需要扩展表示 UNIX 时间戳的二进制位数。

实例 148	计算程序运行时间 光盘位置：光盘\MR\02\148	高级 趣味指数：★★★★

实例说明

在百度中，当执行一个查询操作时，获取到查询结果后页面中就会出现一行文字，提示根据关键字搜索到多少个结果，以及搜索所用的时间。这个时间就是程序在执行该搜索时所用的时间，那么它是如何实现的呢？下面就模仿它来做一个计算程序运行时间的小程序，其运行结果如图 2.123 所示。

距离2016年奥运会开幕还有 450 天!!!

倒计时程序的运行时间为 0.000311036804312 秒

图 2.123　计算程序运行时间

关键技术

本实例的关键点是 strtotime()函数的使用，其原理是：首先在执行查询操作之前定义一个时间，将其精确到微秒，然后执行查询操作，接着在查询功能执行完毕后再次获取一个时间，同样精确到微秒，应用后获取的时间减去最初获取的时间，就是本次查询所用的时间。

设计过程

创建 index.php 文件。首先定义 run_time()函数，获取系统当前的时间戳。然后调用 run_time()方法，获取当前时间戳，接着运行程序。最后，再次调用 run_time()函数，获取系统当前时间，并且输出差值，获取程序的运行时间，其代码如下：

```
<?php
/*	声明 run_time()函数
通过 microtime()函数获取当前时间的微秒数和时间戳
应用 explode()函数将 microtime()函数返回的字符串进行分隔，返回一个数组
包括两个元素，一个元素是当前时间的微秒数，另一个是当前时间的时间戳
应用 list()函数将 explode()函数返回的数组值赋给指定的变量
最后，获取两个变量值的和
*/
function run_time(){
list($msec, $sec) = explode(" ", microtime());                    //使用 explode()函数返回两个变量
return ((float)$msec + (float)$sec);                              //返回两个变量的和
}
$start_time = run_time();                                         //第一次运行 run_time()函数
/*	运行 PHP 代码段	*/
    $time1 = strtotime(date( "Y-m-d"));                           //当前的系统时间
    $time2 = strtotime("2016-08-05");                             //奥运会的开幕时间
    $sub2 = ceil(($time2 - $time1) / 86400);                      //(60 秒*60 分*24 小时)秒/天
```

```
  echo "距离 2016 年奥运会开幕还有<font color=red>$sub2 </font>天!!!";
/*       *******************       */
$end_time = run_time();                                              //再次运行 run_time()函数
?>
<!--   输出差值，获取程序的运行时间  -->
倒计时程序的运行时间为<font color=blue> <?php echo ($end_time - $start_time); ?> </font>秒
?>
```

秘笈心法

心法领悟 148：在 PHP 中应用日期、时间函数时的注意事项。

- ☑ 在配置 PHP 时，进行正确的时区设置，这样就不会在输出时间时出现输出结果与当地时间不符的问题。
- ☑ 在正确地配置了 PHP 的时区之后，就要使用日期、时间函数获取日期、时间或者对应的时间戳，那么必须明确一个日期、时间在不同的操作系统中的有效范围，避免出现获取超出范围的日期、时间数据的现象。
- ☑ 有效的时间戳范围是格林威治时间 1901 年 12 月 13 日 20:45:54~2038 年 1 月 19 日 03:14:07（此范围符合 32 位有符号整数的最小值和最大值）。
- ☑ 在 Linux 系统中此范围限制为 1901 年 12 月 13 日~2038 年 1 月 19 日。
- ☑ 在 Windows 系统中此范围限制为 1970 年 1 月 1 日~2038 年 1 月 19 日。
- ☑ 在通过日期、时间函数获取日期、时间或者时间戳时，如果没有指定时间戳，那么获取的是本地时间。

第 3 章

Web 页面交互

- 获取表单元素的值
- 文本框组件的应用
- 下拉列表的应用
- 表单元素的动态操作
- CSS+ DIV 页面布局
- 弹出窗口
- 状态栏设计
- 导航菜单设计

3.1　获取表单元素的值

获取表单元素提交的值是表单应用中最基本的操作方法，通过获取表单元素提交的值可以实现很多的功能，例如，添加用户注册信息、提交用户登录信息、添加学生详细信息等。下面通过具体实例来介绍获取表单元素提交的值在 PHP 中的应用。

实例 149	设计论坛登录界面 光盘位置：光盘\MR\03\149	高级 趣味指数：★★★★

实例说明

论坛是许多网站不可或缺的模块之一，它为人们相互之间进行各种沟通和交流提供了一个空间。设计论坛模块的第一个步骤就是设计论坛登录界面，让用户根据自己注册的用户名和密码登录论坛。在本例中笔者设计了一个简单的论坛登录界面，效果如图 3.1 所示。



图 3.1　登录界面

关键技术

在设计论坛登录界面时，主要采用表格标签进行界面布局，设计一个 3 行 2 列的表格，如图 3.2 所示。

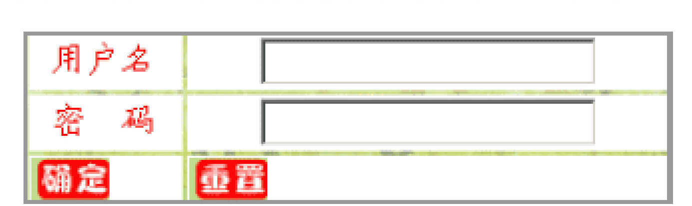


图 3.2　利用表格标签布局

这样，一个论坛登录界面的轮廓就形成了。接下来还需要为表格设置背景图像，利用表格标签的 background 属性设置即可，代码如下：

```
<table width="392px" height="249px" align="center"><tr><td background="pic/tablebk.jpg">
```

设计过程

（1）利用 Dreamweaver 开发工具创建一个动态 PHP 页，将其保存为 index.php。
（2）添加一个表单，在表单中插入一个 3 行 2 列的表格，并调整表格各行的高度和列的宽度。
（3）设置表格的 align 属性为 center，使得表格在表单中始终居中显示。
（4）向表格单元格中输入用户名称和密码信息，并添加文本字段和按钮。
（5）设置“密码”文本字段的类型（type）属性为 password，使其以掩码的形式隐藏真实密码。
（6）实例主要代码如下：

```
<form action="" method="post">
        <table align="center">
                <tr><td>用户名</td><td><input type="text" name="te" size="20" /></td></tr>
                <tr><td>密  码</td><td><input type="password" name="tex" size="22" /></td></tr>
                <tr><td align="left"><input class="two"type="submit" name="sub" value="       " /></td>
                <td align="left"><input class="three" type="reset" name="res" value="       " /></td></tr>
        </table>
</form>
```

秘笈心法

心法领悟 149：密码框与文本字段宽度不一致的解决方案。

许多初学人员都会遇到这种情况，在设计时密码框的宽度与文本字段宽度相同，但是在浏览器中显示则不相同。解决方法是在密码框和文本字段中指定统一的 CSS 样式，即设置 class 属性。在 PHP 页面中添加一个 CSS 样式：

```
<style type="text/css">
.fixwidth{
width : 200px;
…}
</style>
```

这样，问题就解决了。

实例 150 可以上传图片的表单

光盘位置：光盘\MR\03\150

高级

趣味指数：★★★★

实例说明

许多大型网站为了吸引用户，会分配一些空间供用户使用。例如，大家经常使用的电子邮箱、QQ 个人空间等。为了能够让用户在个人空间中放置自己的内容，网站都会提供文件、图片等信息的上传功能。例如在 QQ 个人空间中，用户可以上传自己的照片，供好友欣赏。本实例利用文件域实现图片的上传功能，效果如图 3.3 所示。



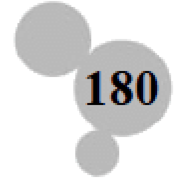
图 3.3 可以上传图片的表单

关键技术

在本例中，实现照片上传的主要方式是利用文件域让用户加载本地文件，然后应用 PHP 函数 move_uploaded_file()实现文件的上传，效果如图 3.4 所示。



图 3.4 文本域加载本地文件



move_uploaded_file()函数可将上传的文件移动到新位置，其语法如下：

```
bool move_uploaded_file ( string filename, string destination );
```

参数说明：

string filename：指定的文件是合法的上传文件。

string destination：定义上传文件的名称。

设计过程

（1）创建动态 PHP 文件，命名为 index.php。

（2）添加一个表单，将 type 属性设置为 file，代码如下：

```
<form action="" method="post" enctype="multipart/form-data">
	<input type="file" name="upfile" size="20" />
	<input class="two"type="submit" name="sub" value="       " />
</form>
```

（3）利用<table>标签编写网页框架。

（4）利用编码实现上传功能，核心代码如下：

```
<?php
if($_POST[sub]){                                                    //单击“上传”按钮
	if($_FILES['upfile']['name'] == ''){                        //判断上传文件名称是否为空
		echo "<script>alert('上传内容为空');</script>";
	}else{
		$info = $_FILES['upfile'];                              //将上传文件信息保存在数组中
		if($info['size'] > 0 && $info['size'] < 1024 * 8000){   //判断上传文件的大小
			$dir = 'upfiles/';                                  //定义变量
			$name = $info['name'];                              //设定文件名称
			$rand = rand(0,10000000);
			$name = $rand.date('YmdHis').$name;
			$path = 'upfiles/'.$name;
			if(!is_dir($dir)){                                  //判断文件夹是否存在，如果不存在则创建一个文件夹
				mkdir($dir);
			}
			$move = move_uploaded_file($info['tmp_name'],$path);   //将上传文件移动到指定位置
			if($move == true){
				echo "<script>alert('上传文件成功');</script>";
			}
		}else{
			echo "<script>alert('上传文件过大');</script>";
		}
	}
}
?>
```

秘笈心法

心法领悟 150：解决上传文件重名问题。

如果用户建立个人网站，允许他人上传文件到服务器中，但是每一个浏览网站的用户并不知道服务器中是否存在与即将上传的文件重名的文件。大家都知道，在 Windows 系统下，同一个文件夹中不允许两个同名文件的存在，这就使得用户上传的文件名必须是独一无二的。笔者为上传文件的名称设置了如下代码：

```
$name = $info['name'];                                              //设定文件名称
$rand = rand(0,10000000);
$name = $rand.date('YmdHis').$name;
$path = 'upfiles/'.$name;
```

在一般情况下，这样可以避免上传文件重名的现象。

实例 151　通过 POST 方法提交表单元素

光盘位置：光盘\MR\03\151

高级　趣味指数：★★★★

实例说明

POST 方法是提交表单元素的方法之一，是程序员提交表单的主要途径。运行本实例，效果如图 3.5 所示。



图 3.5　通过 POST 方法提交表单元素

关键技术

本实例主要是通过定义 form 表单的 method 属性实现，方法如下：

```
<form action="" method="post">
```

设计过程

（1）编写脚本文件 index.php。利用<table>标签建立网页框架，引入 CSS 文件。

（2）编写表单，将提交方法定义为 post，代码如下：

```
<form action="" method="post">                                    //提交方法为 post
        <br>
        用户名：<input class="one" type="text" name="te" size="20" />
        <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value="  " />
        <input class="three" type="reset" name="res" value="  " />
</form>
```

秘笈心法

心法领悟 151：为什么 POST 方法是程序员处理表单程序的首选。

POST 方法与 GET 方法不同，虽然都是采用地址栏传递参数的方式，但是通过 POST 方式传递的参数在地址栏中是不可见的，这样就大大降低了网站或者程序的外来因素的威胁。

实例 152　通过$_POST 方法获取表单元素

光盘位置：光盘\MR\03\152

高级　趣味指数：★★★★

实例说明

实例 151 介绍了如何通过 POST 方法提交表单元素，本实例讲解如何通过$_POST 方法获取表单元素。运行本实例，效果如图 3.6 所示。



图 3.6　通过$_POST 方法获取表单元素

关键技术

本实例主要是利用预定义变量$_POST，该变量是一个自动全局变量，即它在所有的 PHP 脚本中都有效。本实例通过单击按钮实现访问文本框信息的代码如下：

```
<tr>
<td width="100px" align="center"><?php echo $_POST[te];?></td>
<td width="100px" align="center"><?php echo $_POST[pwd];?></td>
</tr>
```

设计过程

（1）编写脚本文件 index.php。通过<table>标签编写网页框架，引入 CSS 文件，建立表单，代码如下：

```
<form action="" method="post">
        <br>
        用户名：<input class="one" type="text" name="te" size="20" />
        <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value="  " />
        <input class="three" type="reset" name="res" value=" " />
</form>
```

（2）当单击“确定”按钮时，程序首先判断文本框和密码框信息是否为空，如不为空，输出通过$_POST 方法获取的数据，核心代码如下：

```
<?php
        if($_POST[sub]){                                                    //单击按钮
                if($_POST[te] == "" || $_POST[pwd] == ""){                  //判断文本框和密码框是否为空
                        echo "<script>alert('文本框不能为空');</script>";
                }else{
?>
<table border="1" bordercolor="#0000CC" cellspacing="0" width="200px">
        <tr>
                <td align="center">用户名</td>
                <td align="center">密码</td>
        </tr>
                        <tr>
                <td width="100px" align="center"><?php echo $_POST[te];?></td>
                <td width="100px" align="center"><?php echo $_POST[pwd];?></td>
        </tr>
</table>
<?php
                }
        }
?>
```

秘笈心法

心法领悟 152：超级全局变量$_POST。

虽然$_POST 是一个全局变量，但是在访问的时候不需要使用关键字 global。

实例 153	通过 GET 方法提交表单元素 光盘位置：光盘\MR\03\153	高级 趣味指数：★★★★

实例说明

GET 方法是提交表单元素的另一种方法，这种方法虽然不是程序员提交表单的首选，但是在日常的程序开发过程中也是很实用的。运行本实例，效果如图 3.7 所示。



图 3.7　通过 GET 方法提交表单元素

关键技术

本实例主要是通过定义 form 表单的 method 属性来实现，定义方法如下：

```
<form action="" method="get">
```

设计过程

（1）编写脚本文件 index.php，利用<table>标签建立网页框架，引入 CSS 文件。

（2）编写表单，将提交方法定义为 get，代码如下：

```
<form action="" method="get">                                    //提交方法为 get
        <br>
        用户名：<input class="one" type="text" name="te" size="20" />
        <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value="  " />
        <input class="three" type="reset" name="res" value="  " />
</form>
```

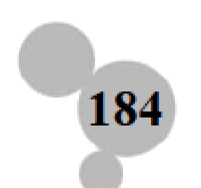
秘笈心法

心法领悟 153：GET 方法与 POST 方法的本质区别。

当用户提交表单时，观察 GET 方法和 POST 方法在地址栏的区别，POST 方法不会有任何变化，而 GET 方法将在地址栏内显示如图 3.8 所示的内容。

http://localhost/MR/03/005&006/in.php?te=mr&pwd=mrsoft&sub=%C2%A0%C2%A0%C2%A0%C2%A0%C2%A0%C2%A0%C2%A

图 3.8　GET 方法提交地址栏变化



虽然传递内容经过一些加密处理，但也近乎是明文地将内容显示出来，是非常不安全的行为。

实例 154　通过$_GET 方法获取表单元素　高级

光盘位置：光盘\MR\03\154　趣味指数：★★★★

实例说明

通过上面实例的学习，相信用户已经掌握了什么是 GET 方法以及如何在表单中使用 GET 方法，下面将通过$_GET 方法向用户演示如何获取表单元素，如图 3.9 所示。



图 3.9　通过$_GET 方法获取表单元素

关键技术

本实例主要应用预定义变量$_GET 实现，本变量是一个自动全局变量，即它在所有的 PHP 脚本中都有效。本实例通过单击“确定”按钮，应用$_GET 方法获取文本框提交的数据，其代码如下：

```
<tr>
<td width="100px" align="center"><?php echo $_GET[te];?></td>
<td width="100px" align="center"><?php echo $_GET[pwd];?></td>
</tr>
```

设计过程

（1）编写脚本文件 index.php。通过<table>标签编写网页框架，引入 CSS 文件，建立表单，代码如下：

```
<form action="in.php" method="get">
        用户名：<input class="one" type="text" name="te" size="20" />
      <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value=" " />
        <input class="three" type="reset" name="res" value=" " />
</form>
```

（2）当单击“确定”按钮时，程序首先判断文本框和密码框信息是否为空，当不为空时，输出通过$_GET 方法获取的数据，核心代码如下：

```
<?php
    if($_GET[sub]){                                          //单击按钮
        if($_GET[te] == "" || $_GET[pwd] == ""){             //判断文本框内容
            echo "<script>alert('文本框不能为空');</script>";
        }else{
?>
<table border="1" bordercolor="#0000CC" cellspacing="0" width="200px"><tr><td align="center">用户名</td><td align="center">密码</td></tr>
<tr><td width="100px" align="center"><?php echo $_GET[te];?></td><td width="100px" align="center"><?php echo
$_GET[pwd];?></td></tr></table>
<?php
        }
    }
?>
```

秘笈心法

心法领悟 154：超级全局变量$_GET。

与全局变量$_POST 一样，$_GET 方法也不需要使用关键字 global 来访问。通过$_GET 方法还可以获取到超链接传递的参数值。

3.2 文本框组件的应用

文本框组件是用来输入任何类型的单行文本、数字或字母数据的文本域。在进行程序设计过程中，经常使用文本框组件来获取用户输入的信息。下面通过具体实例来说明文本框组件在 PHP 中的应用。

实例 155	以文本域的形式显示数据信息 光盘位置：光盘\MR\03\155	高级 趣味指数：★★★★

实例说明

在开发网站的过程中，一般涉及用户注册的网站模块都有一个用户注册的服务条款，个别的服务条款以整个网页进行显示，但是有的网站因服务条款篇幅过长，不适合使用这种方式，最好的解决方法是以文本域的形式来显示，既可节省页面，又美观大方。运行本实例，在文本域中显示用户注册的服务条款，运行效果如图 3.10 所示。

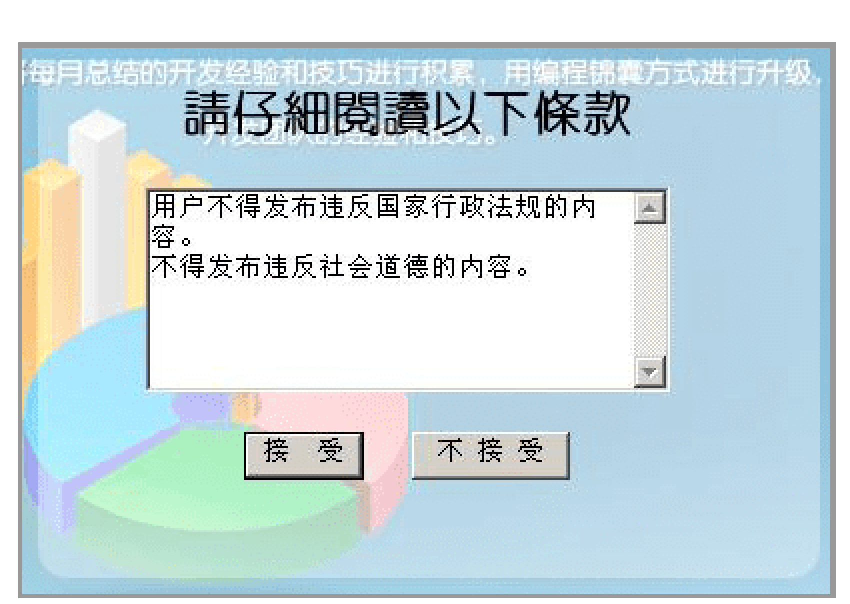


图 3.10 以文本域形式显示数据

关键技术

本实例主要应用表单中的文本域显示信息，主要就是设置文本域的初始值，代码如下：

```
<textarea name="test" cols="45" rows="8" id="test"><?php echo $myrow[text];?></textarea>
```

参数说明：

name：文本域的名称。

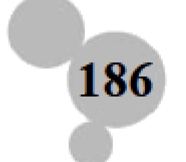
cols：文本域的字符宽度。

rows：文本域显示的行数。

设计过程

编写脚本文件，编写文本域代码，并通过 PHP 变量传递初始值，代码如下：

```
<?php
$str = "用户不得发布违反国家行政法规的内容。                                    //定义变量
不得发布违反社会道德的内容。";
```



```
?>
<form action="" method="post">
<div align="center">
  <textarea name="txt" id="one" cols="30" rows="6"><?php echo $str;?></textarea>              //传递初始值
  <br>
  <br>
    <input type="submit" name="agreen" id="agreen" value="接　受(8)"   disabled="disabled"/>
          
    <input type="submit" name="disagreen" id="disagreen" value="不 接 受" />
    </div>
</form>
```

秘笈心法

心法领悟 155：文本域标签的初始值。

文本框标签和其他一些标签设置初始值的方法是将初始内容写在 value 属性中，而文本域标签的初始值内容是写在文本域的开始和结束标签之间。

实例 156　限制多行文本域输入的字符个数

光盘位置：光盘\MR\03\156

高级

趣味指数：★★★★

实例说明

在网站的论坛或意见箱中，留言的内容通常都是在文本域中输入的，而文本域不具有限制用户输入最多字符数的功能，因此必须由开发人员编写代码进行控制。运行本实例，在“意见内容”文本框中只能输入 1000 个字节，即 1000 个英文字母或 500 个中文汉字，用户在文本框中每输入一个字符，下面的字节位置都将自动计算已输入字节数和剩余字节数，运行结果如图 3.11 所示。

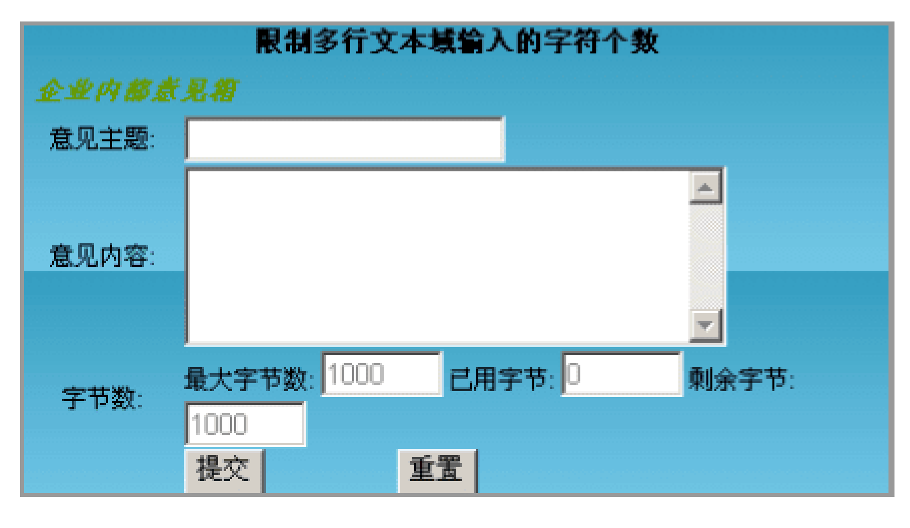


图 3.11　限制文本域输入的字符个数

关键技术

要实现限制多行文本域输入的字符个数，关键是应用 onKeyDown 事件和 onKeyUp 事件，调用自定义的 JavaScript 函数 StatByte()来限制输入的最多字符数并计算已用字节数和剩余字节数。

设计过程

（1）创建一个企业内部意见箱的文本框，代码如下：

```
<textarea name="suggestions" cols="30" rows="5" class="wenbenkuang" id="suggestions"></textarea>
```

（2）添加显示字节数的文本框，包括最大字节数、已用字节和剩余字节，注意此时应该将这 3 个文本框设置为停用状态，不允许用户修改，代码如下：

```
最大字节数:
  <input name="maxs" type="text" disabled class="noborder" id="maxs"   value="1000" size="4">
已用字节:
<input name="used" type="text" disabled class="noborder" id="used"   value="0" size="4">
```

```
剩余字节:
<input name="remain" type="text" disabled class="noborder" id="remain" value="1000" size="4">
```

（3）编写自定义的 JavaScript 函数 StatByte()用于判断用户输入的字符数是否大于允许输入的字符数，并计算已用字节数与剩余字节数，代码如下：

```
<SCRIPT language=JavaScript>
var LastCount =0;
function StatByte(Message,maxs,Used,Remain){                                    //字节统计
  var ByteCount = 0;
  var StrValue   = Message.value;
  var StrLength = Message.value.length;
  var maxsValue   = maxs.value;
if(LastCount != StrLength) {                                                      //在此判断，减少循环次数
 for (i=0;i<StrLength;i++){
        ByteCount   = (StrValue.charCodeAt(i)<=256) ? ByteCount + 1 : ByteCount + 2;
        if (ByteCount>maxsValue) {
              Message.value = StrValue.substring(0,i);
        alert("留言内容最多不能超过 " +maxsValue+ " 个字节！ \n 注意：一个汉字为两字节。");
            ByteCount = maxsValue;
            break;
        }
}
    Used.value = ByteCount;
    Remain.value = maxsValue - ByteCount;
    LastCount = StrLength;
 }
}
</SCRIPT>
```

（4）修改“意见内容”文本域的代码，在 onKeyDown 事件和 onKeyUp 事件中调用自定义的 JavaScript 函数 StatByte()，限制输入的最多字符数并计算已用字节数和剩余字节数，代码如下：

```
<textarea name="suggestions" cols="30" rows="5" class="wenbenkuang" id="suggestions"
        onKeyDown="StatByte(this.form.suggestions,this.form.maxs,this.form.used,this.form.remain);"
        onKeyUp="StatByte(this.form.suggestions,this.form.maxs,this.form.used,this.form.remain);">
</textarea>
```

秘笈心法

心法领悟 156：字节数的计算。

在计算字节数时，需要判断输入的字符是汉字还是英文字母或数字，因为一个汉字占两个字节。

实例 157	设置文本框的只读属性 光盘位置：光盘\MR\03\157	高级 趣味指数：★★★★

实例说明

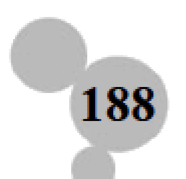
在网页中，有些文本框中的信息是不允许浏览者进行修改的，即用户只有浏览的权限而没有修改的权限。此功能可以通过设置文本框的只读属性来实现。运行本实例，在页面中修改库存数量时，将弹出“此文本框为只读属性，库存数量不能修改！！”的提示框，运行结果如图 3.12 所示。

关键技术

本实例的关键是文本框中 readonly 属性的运用，通过它可设置文本框的只读属性，然后通过 onfocus 事件调用 JavaScript 脚本，当用户单击已经设置为只读属性的文本框时，弹出一个提示对话框。

使用 readonly 属性实现只读的代码如下：

```
<input id="o" class="o" type="text" readonly value='0100521041'/>
```



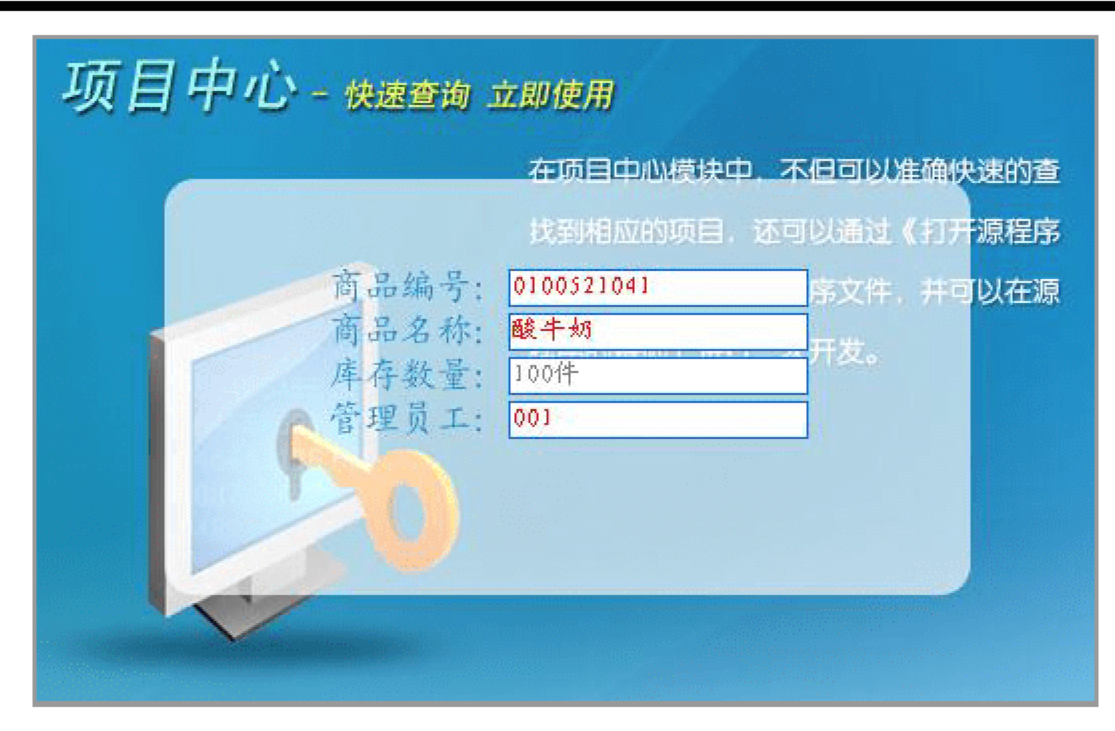


图 3.12　设置文本框的只读属性

设计过程

新建 index.php 文件。编写表格，在表格中插入<form>标签，编写文本框，设置文本框的属性为 readonly，代码如下：

```
<form action="" method="post">
商品编号：<input id="o" class="o" type="text" readonly value='0100521041'/><br>
商品名称：<input id="o"type="text" value="酸牛奶" onfocus="this.value=''" /><br>
库存数量：<input id="o" class="n"type="text" disabled="disabled" value="100 件" /><br>
管理员工：<input id="o"type="text" value="001" />
</form>
```

秘笈心法

心法领悟 157：使用 readonly 的小技巧。

在使用 readonly 时，只需在<input>标记中加入属性名称即可。

实例 158	自动计算金额 光盘位置：光盘\MR\03\158	高级 趣味指数：★★★★

实例说明

在物流管理系统中，经常需要计算出库商品的金额或入库商品的金额，若以手工计算，比较麻烦而且容易出错，如果在程序中实现自动计算金额功能，则可以减少许多不必要的麻烦。运行本实例，效果如图 3.13 所示。



图 3.13　自动计算金额

关键技术

本实例主要通过触发表单控件的 blur 事件，即表单控件失去焦点时，实现自动计算金额。实现此功能的 JavaScript 代码如下：

```
$(".o").blur(function(){                                    //当商品单价文本框失去焦点
        var price = $(".o").val();
        var num = $(".n").val();
        var amount = price * num;
        $(".e").val(amount);
});
```

设计过程

编写脚本文件并命名为 index.php。引入 CSS 样式表文件，导入 JavaScript 脚本。编写表单，当商品单价失去焦点时自动计算金额，核心代码如下：

```
<form action="" method="post" name="form1">
商品编号：<input id="o" type="text" readonly value='酸牛奶'/><br>
商品单价：<input id="o" type="text"class="o" value="0.0"/><br>
库存数量：<input id="o" class="n"type="text" value="35" readonly /><br>
金额总计：<input id="o" class="e"type="text" value="0.0" />
</form>
```

秘笈心法

心法领悟 158：关于 blur 事件。

用于当 blur 事件（即失去焦点事件）发生时执行 JavaScript 代码。适用对象包括 Button、Checkbox、FileUpload、Layer、Password、Radio、Reset、Select、Submit、Text、Textarea 和 Window。

实例 159	为文本框设置默认值 光盘位置：光盘\MR\03\159	高级 趣味指数：★★★★

实例说明

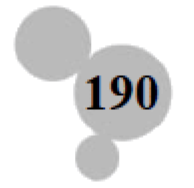
在实例 157 中已经介绍了文本框只读属性的相关知识，下面介绍一种与只读属性类似的属性，即设置文本框默认值。运行本实例，效果如图 3.14 所示，设置文本框的默认颜色为灰色，当在用户名中输入信息时动态改变文本框默认值信息。



图 3.14　为文本框设置默认值

关键技术

本实例的关键是通过 disabled 属性设置文本框的默认值，然后通过相关事件，动态改变默认值文本框的值。



使用 disabled 属性设置默认值的代码如下：

```
<input id="o" class="o" type="text" disabled   value='0100521041'/>
```

设计过程

（1）新建 index.php 文件，引入 CSS 样式表文件，编写表单，代码如下：

```
table width="350" border="0" class="p">
  <tr>
    <td width="100">用户名：</td>
    <td align="left"><input class="m" id="name"type="text" /></td>
  </tr>
  <tr>
    <td>职  位：</td>
    <td align="left"><input class="m"type="text" id="type" disabled value="自动选取"/></td>
  </tr>
</table>
```

（2）定义 JavaScript 脚本方法，动态改变文本框的默认值，代码如下：

```
$("#name").keyup(function(){
        var name = $("#name").val();
        var manager = 'mr';
        if(name == 'mr'){
                $("#type").val("经理");
        }else{
                $("#type").val("程序员");
        }
});
```

秘笈心法

心法领悟 159：关于设置文本框的默认值。

（1）设置文本框的默认值与设置文本框的只读属性在效果上是相同的。

（2）在使用 disabled 时，只需在<input>标记中加入属性名称即可。

实例 160	设置文本框的样式 光盘位置：光盘\MR\03\160	高级 趣味指数：★★★★

实例说明

form 表单的文本框在默认情况下不够美观，使用默认的文本框样式将影响网站的视觉效果。运行本实例，效果如图 3.15 所示，利用 CSS 代码和简单的 JavaScript 代码，使文本框变得生动、有特色。



图 3.15　设置文本框样式

关键技术

本实例首先为文本框定义 CSS 样式，然后通过设置鼠标事件调用 CSS 样式，动态改变文本框的边框颜色。使用 CSS 样式的代码如下：

```
.mouse{
border:1px #FF0000 solid;
background-color:#FFCCFF;
}
```

使用 JavaScript 定义鼠标事件，代码如下：

```
$("#n").mouseover(function(){
        $("#n").css({border:"1px #0099FF solid",background:"white"});
        $("#n").mouseout(function(){
                $("#n").css({border:"1px red solid",background:"#FFCCFF"});
        });
});
$("#na").mouseover(function(){
        $("#na").css({border:"1px #0099FF solid",background:"white"});
        $("#na").mouseout(function(){
                $("#na").css({border:"1px red solid",background:"#FFCCFF"});
        });
});
```

设计过程

（1）新建 index.php 文件，建立网页架构，编写 form 表单，其关键代码如下：

```
<table width="350" border="0" class="p">
  <tr>
    <td width="100">用户名：</td>
    <td align="left"><input type="text" class="mouse" id="n"/></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td align="left"><input type="text" class="mouse" id="na" /></td>
  </tr>
</table>
```

（2）引入 CSS 样式和 JavaScript 脚本，核心代码见关键技术。

秘笈心法

心法领悟 160：引入 CSS 样式。

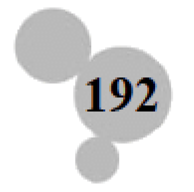
定义 CSS 样式表后需要引入到相关脚本文件中才能起到作用。一般是在<meta>标签下，利用<link>标签引入。以本实例为例，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/inc.css"/>
```

实例 161	文本域的滚动条 光盘位置：光盘\MR\03\161	高级 趣味指数：★★★★

实例说明

文本域本身有自带的滚动条，用户可以通过滚动条浏览文本域中的所有内容，但是在初始情况下，用户可以根据需要设置文本域的列宽，也就是确定文本域在初始情况下显示多少内容。运行本实例，效果如图 3.16 所示。



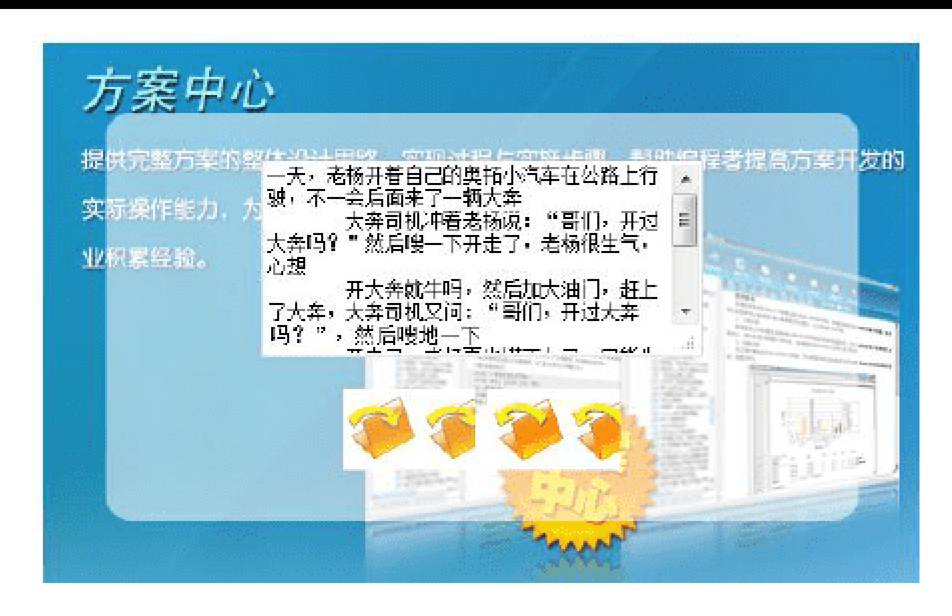


图 3.16　文本域的滚动条

关键技术

本实例的关键点是利用 JavaScript 代码，动态改变文本框的 rows 属性。JavaScript 代码如下：

```
$("#down").click(function(){
        $("textarea").attr("rows","8");
        $("#up").click(function(){
                $("textarea").attr("rows","1");
        });
});
```

设计过程

（1）新建 index.php 文件，通过<table>标签建立网页框架，输入文本域代码，引入 CSS 样式表和 JavaScript 脚本，代码如下：

```
<table width="600" height="600" border="0" align="center" background="pic/bg.jpg">
  <tr>
    <td align="center">
      <textarea cols="40"><?php echo $str;?></textarea><br><br>
      <input   type="button" id='down' value='  '/>
      <input   type="button" id='up' value='  '/>
</td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件，代码见关键技术。

秘笈心法

心法领悟 161：编写 JavaScript 代码的注意事项。

编写 JavaScript 代码时，要注意 JavaScript 代码是区分函数大小写的。

3.3　下拉列表的应用

下拉列表是一种最节省空间的数据显示方式，正常状态下只能看到一个选项，单击控制按钮后，可以显示一定数量的选项，如果超出这个数量，会自动显示滚动条，浏览者可以通过拖动滚动条来查看各选项。

实例 162	省、市级联动菜单 光盘位置：光盘\MR\03\162	高级 趣味指数：★★★★

实例说明

在进行网站开发的过程中，经常会用到相互关联的菜单。例如，省、市、县 3 级联动或者省、市 2 级联动

菜单。在本实例中将讲解如何实现省、市 2 级联动菜单，其运行效果如图 3.17 所示。



图 3.17　省、市级联动菜单

关键技术

本实例通过设置 change 事件实现联动的下拉列表，并且根据 1 级下拉列表传递的不同参数，显示不同的 2 级下拉列表信息。实现这些功能的 JavaScript 代码如下：

```
$("#select1").change( function() {
    var id = $("#select1").val();
    if(id == 1){
        $.get('index.php', null, function(data){
            $("span").empty();
            $("span").append("<select><option>长春</opton><option>松原</opton><option>通辽</opton></select>");
        });
    }else{
        $.get('index.php', null, function(data){
            $("span").empty();
            $("span").append("<select><option>大连</opton><option>旅顺</opton><option>鲅鱼圈</opton></select>");
        });
    }
});
```

设计过程

（1）创建脚本文件，引入 CSS 样式表和 JavaScript 脚本。编写 1 级下拉列表，设置<option>标签属性 value 的值，代码如下：

```
<select id="select1">
    <option value="" selected>请选择</option>
    <option value="1">吉林</option>
    <option value="2">辽宁</option>
</select>
```

（2）编写 JavaScript 代码，当选择 1 级下拉列表项时，触发 change 事件，代码见关键技术。

秘笈心法

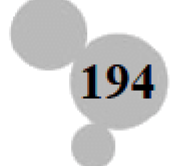
心法领悟 162：JavaScript 脚本语言的小技巧。

如果在 JavaScript 脚本中出现长语句，可以在 JavaScript 语句的最后加上字符“\”，使一条长语句分多行编写。

实例 163	省、市、县级联动菜单 光盘位置：光盘\MR\03\163	高级 趣味指数：★★★★

实例说明

实例 162 中讲解了如何实现 2 级联动菜单，下面将讲解如何实现省、市、县 3 级联动菜单。运行本实例，



效果如图 3.18 所示。



图 3.18　省、市、县级联动菜单

关键技术

实现省、市、县 3 级联动菜单与实现省、市 2 级联动菜单的思路大致相同，但是需要套用多层的 if…else…语句实现。

设计过程

（1）创建脚本文件，命名为 index.php。创建 form 表单，编写下拉列表框，定义 2 级联动和 3 级联动的下拉列表选项，代码如下：

```
<select id="select1">
        <option value="" selected>请选择</option>
        <option value="1">吉林</option>
        <option value="2">辽宁</option>
</select>
<span id='two'></span>
<span id='three'></span>
```

（2）编写 JavaScript 脚本文件，当触发 change 事件时，实现联动效果，代码如下：

```
$("#select1").change( function() {
        var id = $("#select1").val();
        if(id == 1){
                $.get('index.php', null, function(data){
                        $("#two").empty();
                        $("#two").append("<select id='select2'><option>请选择</opton><option value='1'>长春</opton><option value='2'>松原
</opton></select>");
                        $("#select2").change( function() {
                                var page = $("#select2").val();
                                if(page == 1){
                                        $("#three").empty();
                                        $("#three").append("<select><option>双阳</opton><option>农安</opton></select>");
                                }else{
                                        if(page == 2){
                                                $("#three").empty();
                                                $("#three").append("<select><option>乾安</opton><option>大安</opton></select>");
                                        }
                                }
                        });
                });
        }else{
                $.get('index.php', null, function(data){
                        $("#two").empty();
                        $("#two").append("<select id='select3'><option value='1'>大连</opton><option value='2'>旅顺</opton></select>");
                        $("#select3").change( function() {
                                var page = $("#select3").val();
                                if(page == 1){
                                        $("#three").empty();
```

```
                    $("#three").append("<select><option>西岗</opton><option>沙河口</opton></select>");
                }else{
                    if(page == 2){
                        $("#three").empty();
                        $("#three").append("<select><option>金州</opton><option>瓦房店</opton></select>");
                    }
                }
            });
        });
    }
});
```

秘笈心法

心法领悟 163：打造优秀的 3 级联动菜单。

本实例旨在讲述实现 3 级联动菜单的原理，在实际开发中是不建议使用 if…else…语句的，因为这样的多层嵌套方式一方面使程序本身的可读性变差，而且还有可能不利于程序员编程。

实例 164 应用下拉列表框选择所要联机的网站

光盘位置：光盘\MR\03\164

高级

趣味指数：★★★★

实例说明

在开发网站程序的过程中，多数网站都设置了友情链接这一模块，通常情况下，链接的网站以图标、网站名或地址的形式显示在页面中，浏览者可通过单击超链接进入所要访问的网站。为了更好地利用页面，可以通过下拉列表框的方法来创建友情链接。运行本实例，当用户选择下拉列表框中所要联机的网站后（如图 3.19 所示），单击 GO 按钮可以立即链接到指定的网站。

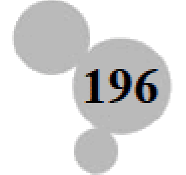


图 3.19　应用下拉列表框选择要联机的网站

关键技术

本实例首先将文本框的 value 属性值定义为所要连接的 IP 地址，然后通过设置鼠标的单击事件实现链接到指定的网站。JavaScript 代码如下：

```
$(document).ready(function(){
$("#button").click(function(){
        var sel = $("#sel").val();
        if(sel != ""){
                $.get(sel);
        }else{
                alert('请选择连接内容');
        }
});
});
```



设计过程

（1）新建 index.php 文件，创建下拉列表框，并添加下拉列表框选项值，代码如下：

```
<select id="sel">
        <option value="" >友情链接</option>
        <option value="http://www.126.com">126 邮箱</option>
        <option value="http://www.163.com">163 邮箱</option>
        <option value="http://www.4399.com">4399 小游戏</option>
    </select>
    <input id="button" class='button' type="submit" value="       " />
```

（2）编写 JavaScript 脚本文件，当单击 GO 按钮时触发事件，完成页面的跳转操作。

秘笈心法

心法领悟 164：使用 CSS 样式表的好处。

利用 CSS 样式可以大量减少网页代码，其原理就是在网页中自定义样式表的选择符，在网页中大量引用这些选择符即可。

实例 165　可输入字符的下拉菜单

光盘位置：光盘\MR\03\165

高级

趣味指数：★★★★

实例说明

通常，网页中应用的下拉列表框的选项值是固定不变的，这样就存在一些弊病，如不能向下拉列表框中添加新的选项值。如果能编写可以输入字符的下拉列表框，就方便很多。运行本实例，在部门的下拉列表框中有“生产部”、“业务部”和“技术部”3 个选项，如果要添加一个“文化部”，直接在下拉列表框中输入“文化部”即可，运行结果如图 3.20 所示。



图 3.20　可输入字符的下拉菜单

关键技术

本实例的开发思路如下：网页中下拉列表的选项值是固定的，要实现在下拉列表框中输入文字，首先要捕获用户的键盘事件，然后根据键盘输入的数据动态添加下拉列表框的选项，同时还要判断用户按下的是 Backspace 键还是 Space 键，如果是 Backspace 键，则需要从光标指针位置向前删除一个字符，如果是 Space 键，则需要在光标指针位置添加一个空格字符。

另外，本实例还利用 onKeyPress 和 onKeyUp 事件调用 JavaScript 自定义函数 press(opt)和 up(opt)。

设计过程

（1）在页面中创建部门下拉列表框，其中包括“生产部”、“业务部”和“技术部”3 个选项，代码如下：

```
<form name="form1" method="post" action="index.php" onKeyPress="press(3)" onKeyUp="up(3)">
        <select name="userclass" size="1" id="userclass">
          <option value="生产部" selected>生产部</option>
          <option value="业务部">业务部</option>
          <option value="技术部">技术部</option>
        </select>
  </form>
```

（2）编写键盘按下事件所执行的 JavaScript 自定义函数 press(opt)，用于在下拉列表框中添加新的选项，并将用户输入的信息添加到下拉列表框中，设置该选项为选中项。

然后编写键盘按下事件所执行的 JavaScript 自定义函数 up(opt)，当用户按 Backspace 键时，将已经写到下拉列表框中的文字逐个删除；当用户按 Space 键时，在下拉列表框中添加一个空格字符，代码如下：

```
function press(opt){
//opt 表示现有可选项的数目
form1.userclass.options[opt]=new Option(ok=(form1.userclass.options[opt])?
form1.userclass.options[opt].innerText+String.fromCharCode(event.keyCode):
String.fromCharCode(event.keyCode),ok)
form1.userclass.selectedIndex=opt;
}
function up(opt){
//opt 表示现有可选项的数目
if(form1.userclass.options[opt]){
        if(event.keyCode==8){
                var str=form1.userclass.options[opt].innerText;
                var len=str.length;
                form1.userclass.options[opt].innerText=str.substring(0,len-1);
                if(form1.userclass.options[opt].innerText==" ")userclass.remove(2);
        }
        if(event.keyCode==32){
                form1.userclass.options[opt].innerText+=" ";
        }
}
}
```

秘笈心法

心法领悟 165：使用 JavaScript 代码删除输入文字的注意事项。

在删除输入选项中的文字时，如果删除到最后一个字符，则将该选项一同删除，同时改变下拉列表框的选中项。

实例 166 设置下拉列表框的默认值

光盘位置：光盘\MR\03\166

高级

趣味指数：★★★★

实例说明

在注册登录模块中，下拉列表框往往被应用在选择用户性别上，如果用户忘记选择此下拉列表框中的内容，就很容易把注册信息弄错。解决这个问题的方法是为下拉列表框设置默认值，如图 3.21 所示，当没有对下拉列表框进行操作时，将弹出提示信息。

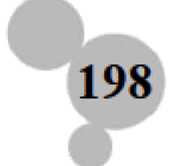




图 3.21　设置下拉列表框的默认值

关键技术

本实例首先为下拉列表框设置默认值，然后通过设置按钮的单击事件实现当用户单击“提交”按钮时，判断下拉列表框的内容是否被选择。

使用 selected 属性实现设置默认值的代码如下：

```
<option selected   value="1">请选择性别</option>
```

设计过程

（1）新建 index.php 文件，引入 CSS 文件和 JavaScript 脚本，编写下拉列表框，代码如下：

```
<select>
  <option selected   value="1">请选择性别</option>
      <option >男性</option>
      <option >女性</option>
</select>
<input id="one" type="button" value="        " />
```

（2）编写 JavaScript 代码，当按钮被单击时触发事件，其代码如下：

```
$(document).ready(function(){
$("#one").click(function(){
        var value = $("select").val();
        if(value == 1){
                alert("您没有选择性别");
        }
});
});
```

秘笈心法

心法领悟 166：关于设置下拉列表框的默认值。

在使用 selected 时，只需在<option>标记中加入属性名称即可，其代码如下：

```
<option selected="selected"></option>
```

实例 167	设置下拉列表框的样式 光盘位置：光盘\MR\03\167	高级 趣味指数：★★★★

实例说明

对页面的美化处理并不属于程序员的工作范畴，而是属于前台美工的分内之事，但是像这种美化下拉列表框的小事情，程序员如果还要和美工去协调未免有点浪费时间，所以学习一些基本的页面美化方法还是很有必

要的。运行本实例，效果如图 3.22 所示。



图 3.22　设置下拉列表框的样式

关键技术

本实例运用 CSS 样式表改变下拉列表框的样式，代码如下：

```
select{
width:100px;
font-family:"楷体_GB2312";
font-size:14px;
color:#FFFFFF;
background-color:#000000;
}
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式，编写下拉列表框，代码如下：

```
<select>
  <option selected   value="1">请选择性别</option>
  <option >男性</option>
  <option >女性</option>
</select>
```

（2）定义 CSS 样式，代码如关键技术中所示。

秘笈心法

心法领悟 167：关于设置下拉列表框的样式。

支持下拉列表框的 CSS 样式很少，所以想设置更多的样式相对来说会很麻烦。如果非要定义一个漂亮的下拉列表样式，最好利用 JavaScript 代码自行编写。

实例 168	应用下拉列表框打开窗口 光盘位置：光盘\MR\03\168	高级 趣味指数：★★★★

实例说明

实例 164 中已经讲解了如何跳转到链接网站的方法，本实例与实例 164 在实现思路上是相同的，唯一不同的是在下拉列表框里通过打开新窗口实现跳转。运行本实例，效果如图 3.23 所示。

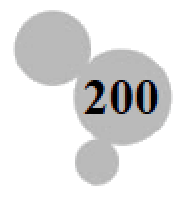




图 3.23　选择链接网站

关键技术

本实例实现的方法是定义 JavaScript 脚本的 change 事件，通过 JavaScript 的 window.open()方法，实现新页面的跳转，代码如下：

```
$(document).ready(function(){
$("#sel").change(function(){
        window.open($("#sel").val());
});
});
```

设计过程

（1）创建脚本文件，建立网页框架，引入 CSS 样式表和 JavaScript 脚本，编写下拉列表，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/inc.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<select id="sel" >
        <option>友情链接</option>
        <option value='http://www.126.com'>126 邮箱</option>
        <option value='http://www.163.com'>163 邮箱</option>
        <option value='http://www.4399.com'>4399 小游戏</option>
</select>
```

（2）编写 JavaScript 脚本文件，当发生 change 事件时发生跳转。

秘笈心法

心法领悟 168：关于 JavaScript。

本实例主要采用 JavaScript 语言和 jQuery 框架编写，jQuery 框架是以 JavaScript 语言编写的，所以两者可以混编。

实例 169	Tab 键在文本域中的体现 光盘位置：光盘\MR\03\169	高级 趣味指数：★★★★

实例说明

Tab 键在编写 Word 等文档时是经常被用到的。在网页的文本域中，按 Tab 键是进行下一个切换项的指示，如何在文本域中体现 Tab 键呢？本实例将向用户讲解。运行本实例，效果如图 3.24 所示。



图 3.24　文本域中体现 Tab 键

关键技术

本实例主要是通过 JavaScript 代码定义 onkeydown 事件实现的，代码如下：

```
function editTab(){
    var code, sel, tmp, r
    var tabs=""
    event.returnValue = false
    sel =event.srcElement.document.selection.createRange()
    r = event.srcElement.createTextRange()
    switch (event.keyCode){
        case (8):
            if (!(sel.getClientRects().length > 1)){
                event.returnValue = true
                return
            }
            code = sel.text
            tmp = sel.duplicate()
            tmp.moveToPoint(r.getBoundingClientRect().left, sel.getClientRects()[0].top)
            sel.setEndPoint("startToStart", tmp)
            sel.text = sel.text.replace(/^\t/gm, "")
            code = code.replace(/^\t/gm, "").replace(/\r\n/g, "\r")
            r.findText(code)
            r.select()
            break
        case (9):
            if (sel.getClientRects().length > 1){
                code = sel.text
                tmp = sel.duplicate()
                tmp.moveToPoint(r.getBoundingClientRect().left, sel.getClientRects()[0].top)
                sel.setEndPoint("startToStart", tmp)
                sel.text = "\t"+sel.text.replace(/\r\n/g, "\r\t")
                code = code.replace(/\r\n/g, "\r\t")
                r.findText(code)
                r.select()
            } else {
                sel.text = "\t"
                sel.select()
            }
            break
        case (13):
            tmp = sel.duplicate()
            tmp.moveToPoint(r.getBoundingClientRect().left, sel.getClientRects()[0].top)
            tmp.setEndPoint("endToEnd", sel)
            for (var i=0; tmp.text.match(/^[\t]+/g) && i<tmp.text.match(/^[\t]+/g)[0].length; i++)     tabs += "\t"
            sel.text = "\r\n"+tabs
            sel.select()
            break
        default:
```

```
                event.returnValue = true
                break
        }
}
```

设计过程

（1）创建脚本文件，引入 JavaScript 脚本，编写文本域标签，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/inc.css"/>
<script type="text/javascript" src="js/in.js"></script>
<textarea cols="30" name="text" rows="3" onkeydown="editTab()"></textarea>
```

（2）当按下键盘按键时触发事件，代码见关键技术。

秘笈心法

心法领悟 169：JavaScript 脚本的引用。

本实例调用 JavaScript 中的 editTab()函数完成对 Tab 键的控制，在页面中引用 JavaScript 文件的方法是应用<script>标签，其代码如下：

```
<script type="text/javascript" src="js/in.js"></script>
```

3.4　表单元素的动态操作

所谓动态操作表单元素是指页面中的表单元素并不是固定不变的，而是在程序执行时由系统根据实际情况自动生成。这里将通过实际的例子来介绍如何动态生成表单元素。

实例 170	投票系统中单选按钮与复选框的应用 光盘位置：光盘\MR\03\170	高级 趣味指数：★★★★

实例说明

在开发程序的过程中，经常应用到在线投票模块，其使用非常简单，只要选中一个单选按钮或者多个复选框所对应的投票主题后，单击“提交”按钮即可完成投票。本实例将介绍投票系统的操作原理，当选中一个投票主题之后，就会弹出相应的投票选区，在该选区中可以添加投票并可以看到投票的结果，运行结果如图 3.25 所示。



图 3.25　在线统计

关键技术

本实例主要是通过文件操作和正则表达式函数实现在线统计程序。其中文件操作的核心代码如下：

```
$file = file_get_contents('info.txt');
	if($file == ""){
		$openn = fopen("info.txt",'a');
		fwrite($openn,'a');
		fclose($openn);
	}
	if(preg_match("/$ip/",$file)){
		echo "<script>alert('同一个 IP 地址，不能投票两次');</script>";
		exit;
	}else{
		if(!$open = fopen("info.txt",'a')){
			print "文件不存在，请创建";
			exit;
		}else{
			fwrite($open,$ip);
			fclose($open);
		}
	}
```

设计过程

（1）创建脚本文件，利用<table>标签建立网页框架，引入 CSS 文件和 JavaScript 脚本，编写单选按钮，代码如下：

```
<form action="" method="post">
	<tr>
		<td class="three"><input id="id" name="check" checked type="radio" value="1" />上网</td>
		<td><img src="pic/logo1.JPG" /><font color='red' size="-1">10 人</font></td>
	</tr>
	<tr>
		<td class="three"><input id="id"name="check" type="radio" value="2" />去娱乐场所</td>
		<td><img src="pic/logo2.JPG" /><font color='red'size="-1">15 人</font></td>
	</tr>
	<tr>
		<td class="three"><input id="id"name="check" type="radio" value="3" />带着女友逛街</td>
		<td><img src="pic/logo3.JPG" /><font color='red'size="-1">20 人</font></td>
	</tr>
	<tr>
		<td class="three"><input id="id"name="check" type="radio" value="4" />睡觉</td>
		<td><img src="pic/logo4.JPG" /><font color='red'size="-1">5 人</font></td>
	</tr>
</form>
```

（2）当单击“确定”按钮时，首先获取客户端 IP 地址，然后将数据写入文本文件中，最后根据提交的值通过 switch 语句进行判断并给出提示信息，代码如下：

```
<?php
if($_POST[sub]){
	$ip = $_SERVER[REMOTE_ADDR];					//取得客户端 IP 地址
	$file = file_get_contents('info.txt');				//将文件内容读取到字符串
	if($file == ""){						//如果文件内容为空
		$openn = fopen("info.txt",'a');				//打开 info.txt 文件
		fwrite($openn,'a');					//向文件中写入一个字符
		fclose($openn);						//关闭文件
	}
	if(preg_match("/$ip/",$file)){					//匹配字符串
		echo "<script>alert('同一个 IP 地址，不能投票两次');</script>";
		exit;							//退出
	}else{
		if(!$open = fopen("info.txt",'a')){
			print "文件不存在，请创建";
```



```
                exit;
            }else{
                fwrite($open,$ip);
                fclose($open);
            }
        }
        switch($_POST[check]){                              //switch 选择语句
            case "2":
            echo "<script>alert('您选择去娱乐场所');</script>";
            break;
            case "3":
            echo "<script>alert('您选择和女友逛街');</script>";
            break;
            case "4":
            echo "<script>alert('您选择睡觉');</script>";
            break;
            case "1":
            echo "<script>alert('您选择上网');</script>";
            break;
        }
}
?>
```

秘笈心法

心法领悟 170：关于对文件的操作。

文件的操作对于所有学习 PHP 这门语言的人来说都会感到头疼，特别是对文件高级操作中的指针的理解，更是如此。所以笔者建议，在学习这部分知识时，一定要先打好基础，然后多实践、多思考。

实例 171　**选中单选按钮后显示其他表单元素**　高级

光盘位置：光盘\MR\03\171　趣味指数：★★★★

实例说明

论坛系统中的会员有不同的级别，一种是超级会员，拥有对论坛的管理权限；另一种是普通会员，只能发布和回复论坛信息。因此在论坛的权限设置模块中就需要应用单选按钮对超级会员和普通会员设置不同的权限。在本实例中，选中“超级用户”单选按钮时，其下方将列出网站的所有管理模块，管理员可以为超级会员设置管理权限，运行结果如图 3.26 所示；选中“普通用户”单选按钮，运行结果如图 3.27 所示。



图 3.26　超级用户设置



图 3.27　普通用户设置

关键技术

选中单选按钮后显示其他表单元素的功能，主要利用 JavaScript 设置表格的 display 样式来实现表格的显示和隐藏。通过设置表格的 ID 属性，可以实现对表格的整体控制，如设置表格的 style（样式）属性。通过将表格的 display 样式设置为 none，可以控制表格的隐藏；将表格的 display 样式设置为 block，可以显示表格。

设计过程

（1）创建脚本文件，利用<table>标签编写网页架构，引入 CSS 样式表和 JavaScript 脚本，编写 form 表单，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table align="center">
  <tr>
    <td>用户名：</td>
    <td><input type="text"/></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td><input type="text"/></td>
  </tr>
  <tr>
    <td>权限设置：</td>
    <td><input id="va" type="radio" name="ch" value="1" checked/>超级用户<input id="val" name="ch"type="radio" value="2" />普通用户</td>
  </tr>
  <tr>
        <td><table id="name1"><tr><td><input type="checkbox" />帖子管理</td></tr></table></td>
        <td><table id="name2"><tr><td><tr><td><input type="checkbox" />后台管理</td></tr></table></td>
  </tr>
  <tr>
        <td><table id="name3"><tr><td><tr><td><input type="checkbox" />数库管理</td></tr></table></td>
        <td><table id="name4"><tr><td><tr><td><input type="checkbox" />公告管理</td></tr></table></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件，将 style 属性动态添加到复选框的表格中，控制表格的隐藏和显示，代码如下：

```
$(document).ready(function(){
 $("#va").click(function(){
        $("#name1").css("display","block");
        $("#name2").css("display","block");
        $("#name3").css("display","block");
        $("#name4").css("display","block");
 });
 $("#val").click(function(){
        $("#name1").css("display","none");
        $("#name2").css("display","none");
        $("#name3").css("display","none");
        $("#name4").css("display","none");
 });
});
```

秘笈心法

心法领悟 171：添加 CSS 属性。

利用 jQuery 框架可以很容易地将属性和值动态地添加到指定标签中，语法格式如下：

```
$("class/id").attr("key","value");
```



实例 172　不提交表单获取单选按钮的值

光盘位置：光盘\MR\03\172

高级

趣味指数：★★★★

实例说明

PHP 是一种服务器端的编程语言，顾名思义，在需要时与服务器进行交互并且处理信息，但是很多时候，用户希望在与服务器交互之前就完成一些操作，这个时候客户端脚本语言 JavaScript 的作用就体现出来了。运行本实例，通过 JavaScript 代码实现不提交表单获取单选按钮的值，如图 3.28 所示。

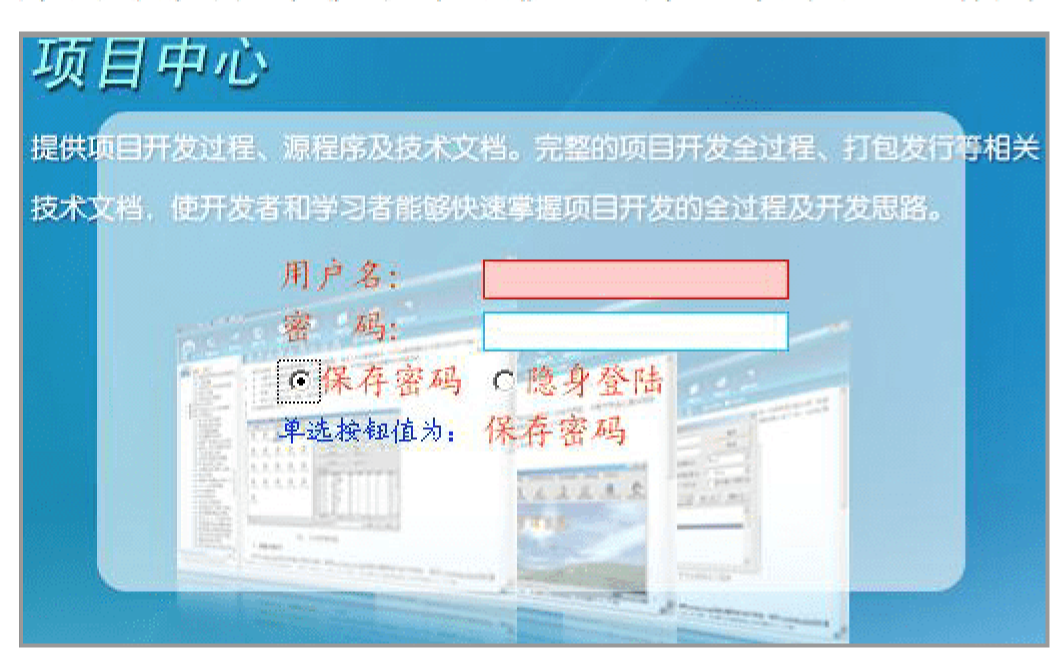


图 3.28　不提交表单获取单选按钮的值

关键技术

本实例主要是通过 JavaScript 代码定义 click 事件，获取单选按钮的值，并且将单选按钮的值通过<div>标签返回到页面中，代码如下：

```
$(document).ready(function(){
$(".n").mouseover(function(){
        $(".n").css({background:"#FFCCCC",border:"1px red solid"});
        $(".n").mouseout(function(){
                $(".n").css({background:"white",border:"1px #00CCFF solid"});
        });
});
$(".o").mouseover(function(){
        $(".o").css({background:"#FFCCCC",border:"1px red solid"});
        $(".o").mouseout(function(){
                $(".o").css({background:"white",border:"1px #00CCFF solid"});
        });
});
$("#e").click(function(){
        $("#nam").css("display","block");
        var value = $("#e").val();

        $("#name").html(value);
});
$("#ee").click(function(){
        $("#nam").css("display","block");
        var value = $("#ee").val();

        $("#name").html(value);
});
});
```

设计过程

（1）创建脚本文件，引入 CSS 样式和 JavaScript 脚本文件，创建用户登录的 form 表单，设置文本框和单

选按钮元素，并且设置<div>标签定义标签的 ID 值，核心代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table align="center">
  <tr>
    <td>用户名：</td>
    <td><input class="n" type="text" /></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td><input class="o" type="text" /></td>
  </tr>
  <tr>
    <td><input id="e" name="ch"type="radio" value="保存密码" />保存密码</td>
    <td><input id="ee" name="ch"type="radio" value="隐身登录" />隐身登录</td>
  </tr>
   <tr>
    <td id="nam">单选按钮值为：</td>
    <td><div id="name"></div></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，引入 jQuery 库文件 jquery-1.3.2.js。

秘笈心法

心法领悟 172：jQuery 中的 click 事件。

当单击标签时触发此事件，语法格式如下：

```
$("class/id").click(function(){…});
```

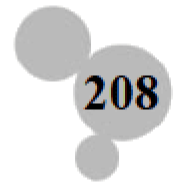
实例 173	实现复选框中的全选、反选和不选 光盘位置：光盘\MR\03\173	高级 趣味指数：★★★★

实例说明

复选框应用在用户注册系统中，如个人爱好选择等。本实例所讲述的 3 个操作：全选、反选、不选，只是根据实际情况，对复选框实现的快捷操作。运行本实例，全选的效果如图 3.29 所示。



图 3.29　实现全选



关键技术

复选框的全选、反选和不选主要应用 JavaScript 脚本，通过自定义函数完成。

（1）在通过 JavaScript 脚本中的自定义函数完成复选框的全选、反选和不选功能中，应用的第 1 个方法是 getElementsByTagName，获取指定标签的名称，返回值为一个包含标签信息的 object。

（2）根据 getElementsByTagName 标签返回的对象，判断标签类型（type）的值是不是 checkbox。

（3）当标签类型 type 的值为 checkbox 时，为标签中的 checked 赋值。当 checked 的值为 TRUE 时，将其赋值为 FALSE；当 checked 的值为 FALSE 时，将其赋值为 TRUE。

（4）在页面中通过 script 标签调用 js 文件夹中的文件，应用 onClick 事件调用自定义函数，完成全选、反选和不选功能的实现。

设计过程

（1）创建 js 文件夹，编写 reg.js 脚本文件。在 reg.js 文件中，编写自定义函数，实现全选、反选和不选功能。reg.js 文件的关键代码如下：

```
function uncheckAll(form1,status) {                          //不选
 var elements = form1.getElementsByTagName('input');         //获取 input 标签
 for(var i=0; i<elements.length; i++){                       //根据标签的长度执行循环
        if(elements[i].type == 'checkbox') {                 //判断对象中元素的类型
                if(elements[i].checked==true){               //判断当 checked 的值为 TRUE 时
                        elements[i].checked=false;           //将 checked 赋值为 FALSE
                }
        }
 }
}
function checkAll(form1,status){                             //全选
 var elements = form1.getElementsByTagName('input');
 for(var i=0; i<elements.length; i++){
        if(elements[i].type == 'checkbox') {
                if(elements[i].checked==false){
                        elements[i].checked=true;
                }
        }
 }
}
function switchAll(form1,status) {                           //反选
 var elements = form1.getElementsByTagName('input');
 for(var i=0; i<elements.length; i++){
        if(elements[i].type == 'checkbox'){
                if(elements[i].checked==true){
                        elements[i].checked=false;
                }else if(elements[i].checked==false){
                        elements[i].checked=true;
                }
        }
 }
}
```

（2）创建 index.php 页面，输出会员信息，并添加图像按钮，通过 onClick 事件调用 JavaScript 自定义函数实现全选、反选、不选和删除的功能，其关键代码如下：

```
<script language="javascript" src="js/reg.js"></script>
<form method="post" name="form1" id="form1" action="index_ok.php">
<tr>
    <td width="62" align="center"><input name="conn_id[]" type="checkbox" id="conn_id[]" value="1" /></td>
    <td width="242">欧阳</td>
    <td width="243">PHP</td>
    <td width="485">部门经理</td>
    <td width="485">29</td>
```

```
</tr>
<tr>
<td colspan="5" align="center"><!--通过 onClick 事件调用自定义函数，执行相应的操作-->
    <img src="images/bg_19-20.jpg" onclick="checkAll(form1,status)" width="62" height="25" />
    <img src="images/bg_14-14.jpg" onclick="switchAll(form1,status)"   width="62" height="25" />
    <img src="images/bg_07-08.jpg" width="62" height="25" onclick="uncheckAll(form1,status)" />
    <input type="image" name="imageField" src="images/bg_14.jpg" />
</td>
</tr>
</form>
```

秘笈心法

心法领悟 173：复选框的全选、反选和不选的实现思路。

实现复选框的全选、反选和不选操作主要应用 JavaScript 脚本，通过 JavaScript 脚本中的自定义函数，判断复选框中 checked 的值。全选就是将 checked 赋值为 TRUE，不选则是将 checked 赋值为 FALSE，而反选则是当 checked 的值为 TRUE 时，将其赋值为 FALSE；当 checked 的值为 FALSE 时，将其赋值为 TRUE。

实例 174	隐藏域提交用户的 ID 值 光盘位置：光盘\MR\03\174	高级 趣味指数：★★★★

实例说明

开发在线论坛系统时，在回复帖子的模块中，要记录是谁回复了哪一个帖子，即要把这个回复用户的 ID 同回复帖子的内容一起保存起来。回帖的内容是从文本域中获取的，而回帖人的 ID 则是通过隐藏域进行提交的，不需要手动填写，如图 3.30 所示。

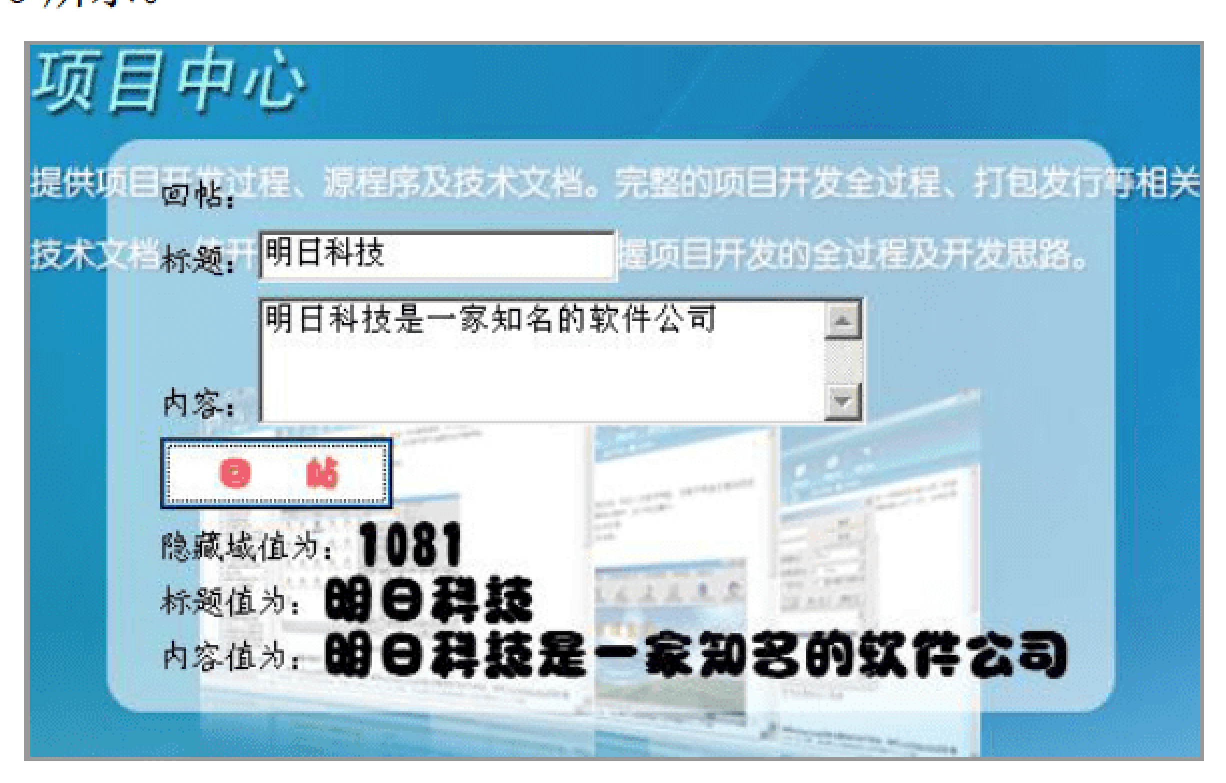


图 3.30　隐藏域提交用户 ID

关键技术

本实例主要应用表单中的隐藏域来实现文件值的提交和获取。

表单中隐藏域的语法格式如下：

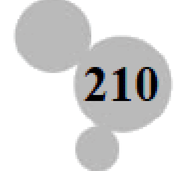
```
<input type= "hidden" name="ID" value="">
```

参数说明：

type：表单的类型，其中的 hidden 表示隐藏域。

name：隐藏域的名称，可以自己定义。

value：隐藏域的值，可以填写隐藏域的默认值。



设计过程

（1）创建脚本文件，利用<table>标签建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table width="600px" height="600px" background="pic/bg.jpg">
  <tr>
    <td>
      <table align="center">
  <tr>
    <td>回帖：<input type="hidden" name="ID" value="1081" /></td>
  </tr>
  <tr>
    <td>标题：<input type="text"   /></td>
  </tr>
  <tr>
    <td>内容：<textarea cols="30" rows="3"></textarea></td>
  </tr>
  <tr>
    <td><input class="one" type="submit" name="sub" value="" /></td>
  </tr>
  <tr><td id="name" style="display:none">
  隐藏域值为：<b id="b1"></b><br>标题值为：<b id="b2"></b><br>内容值为：<b id="b3"></b>
  </td></tr>
</table>
 </td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本，当单击“回帖”按钮时触发事件，代码如下：

```
$(document).ready(function(){
 $(".one").click(function(){
        var val1 = $("input:eq(0)").val();
        var val2 = $("input:eq(1)").val();
        var val3 = $("textarea").val();
        $("#name").css("display","block");
        $("#b1").html(val1);
        $("#b2").html(val2);
        $("#b3").html(val3);
 });
});
```

秘笈心法

心法领悟 174：jQuery 中使用$("").html()方法的注意事项。

此方法是取得第 1 个匹配元素的 HTML 内容。这个方法不能用于 XML 文档，但可以用于 XHTML 文档。

实例 175	图像域替代提交按钮 光盘位置：光盘\MR\03\175	高级 趣味指数：★★★★

实例说明

在很多网站的主页布局中，程序员除了在导航栏中布置一些按钮外，在网页的主体部分中很少出现类似于提交的按钮，这样做的主要目的并不是因为网页中不再需要按钮，而是被一些超链接或者图像域跳转所替代。运行本实例，当单击图片时实现表单提交，如图 3.31 所示。

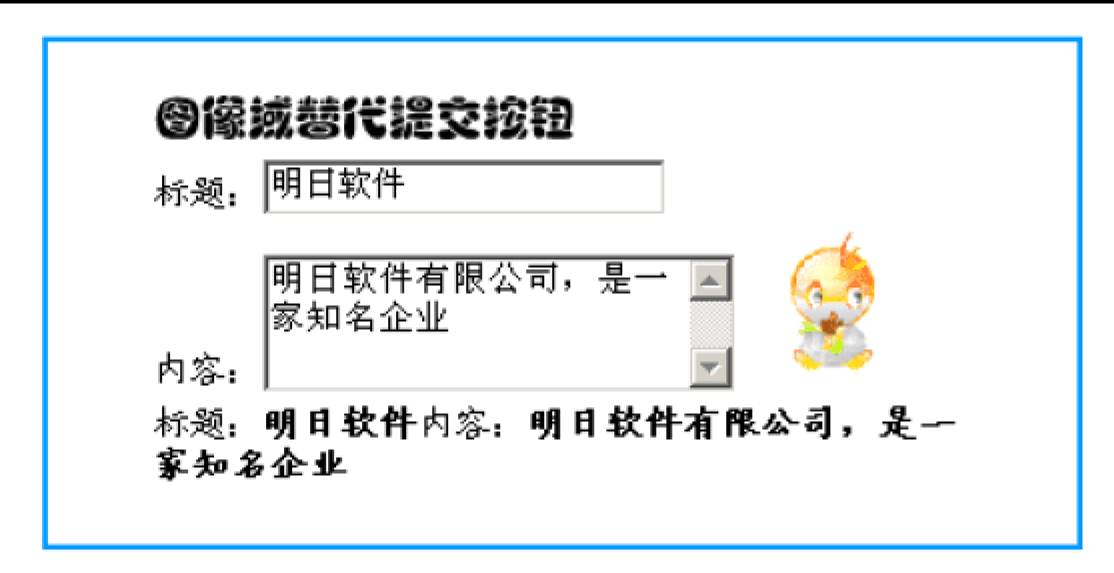


图 3.31　图像域替代提交按钮

关键技术

本实例主要是利用 JavaScript 代码，通过单击事件完成数据的提交操作，其关键代码如下：

```
$(document).ready(function(){
$("img").click(function(){
        var input = $("input").val();
        var textarea = $("textarea").val();
        if(input == "" || textarea == ""){
                alert("文本框为空");
        }else{
                $(".n").css("display","block");
                $("#i").html(input);
                $("#t").html(textarea);
        }
});
});
```

设计过程

（1）创建脚本文件，建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下：

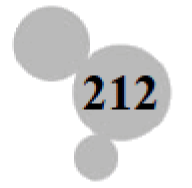
```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table width="319" align="center">
  <tr>
    <td width="296" class="oo">图像域替代提交按钮</td>
  </tr>
  <tr>
    <td>标题：<input type="text" /></td>
  </tr>
  <tr>
    <td>内容：<textarea cols="20" rows="3"></textarea>  <img src="pic/img.gif" /></td>
  </tr>
  <tr>
    <td class="n">标题：<b id="i"></b>内容：<b id="t"></b></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，当单击图片时触发事件，代码如关键技术所示。

秘笈心法

心法领悟 175：图像域的使用。

使用图像域进行跳转和提交的方式有很多。例如嵌入到<a>标签中，进行客户端跳转；或者应用在 form 表单的提交、重置按钮中。灵活运用图像域的知识会使程序更美观。



实例 176　跳转菜单实现页面跳转

光盘位置：光盘\MR\03\176

高级

趣味指数：★★★★

实例说明

跳转菜单在网站中是很常见的，一般被放置在靠近头部的位置，又称导航栏。下面就讲解如何实现单击按钮页面跳转的效果，如图 3.32 所示。



图 3.32　跳转菜单实现页面跳转

关键技术

本实例主要通过<map>定义跳转，代码如下：

```
<map name="Map2" id="Map2">
  <area shape="rect" coords="89,64,141,94" href="index.php?id=1" />
  <area shape="rect" coords="168,64,229,94" href="index.php?id=2" />
  <area shape="rect" coords="250,65,305,99" href="index.php?id=3" />
  <area shape="rect" coords="328,63,388,97" href="index.php?id=4" />
  <area shape="rect" coords="13,65,61,93" href="index.php" />
</map>
```

设计过程

（1）创建 index.php 文件，设计网页，通过<map>定义页面跳转的超链接。

（2）编写 PHP 代码，当单击不同按钮时实现跳转，代码如下：

```
<?php
	switch($_GET['id']){
		case "1":
		echo "pic/reg.JPG";
		break;
		case "2":
		echo "pic/log.JPG";
		break;
```

```
            case "3":
            echo "pic/for.JPG";
            break;
            case "4":
            echo "pic/mes.JPG";
            break;
            default:
            echo "pic/for.JPG";
            break;
        }
?>
```

秘笈心法

心法领悟 176：如何取得地址栏参数。

以本实例为例子，地址组成格式如下所示，可以通过$_GET 方式取得相应的参数并进行操作。

```
http://地址/主页.php?参数&参数
```

实例 177 上传图片预览

光盘位置：光盘\MR\03\177

高级

趣味指数：★★★★

实例说明

上传图片时最好为程序设置预览功能，防止用户意外传错图片。还有一个很重要的模块就是获取图片的地址，方便用户网站的推广。运行本实例，完成预览图片和获取图片地址的功能，运行结果如图 3.33 所示。



图 3.33　上传预览图片

关键技术

本实例通过 JavaScript 的 click 事件，实现图片预览、获取地址和清除上传文本框信息的功能，其核心代码如下：

```
$(document).ready(function(){
 $("#two").click(function(){                                     //单击预览图片
        var value = $("#one").val();                             //取得上传地址
        if(value == ""){                                         //判断地址是否为空
                alert("地址为空");
        }else{
                $(".n").append("<img id='five' src="+value+">");  //显示图片
                $("#two").click(function(){
                        $(".n").empty();
                        $(".n").append("<img id='five' src="+value+">");
```



```
                });
            }
    });
    $("#three").click(function(){                                    //显示图片地址
        var value = $("#one").val();
        if(value == ""){
            alert("地址为空");
        }else{
            alert(value);
        }
    });
    $("#four").click(function(){                                     //清除文本框信息
        $(".e").empty();
        $(".e").append("<input class='one' id='one' type='file'>");
    });
});
```

设计过程

（1）新建 index.php 文件，引入 CSS 样式、in.js 文件以及 jQuery 库文件 jquery-1.3.2.js。创建 form 表单，设置文件域属性，添加预览图片、图片地址和清除的按钮，代码如下：

```
<table width="600" height="450" align="center" background="pic/bg.jpg">
  <tr>
    <td align="center"><div class="n"></div>
      <table align="center">
        <tr>
            <td>
            <span class="o">
            <span class="e"><input class="one" id="one" type="file"></span>
            <input class="one" id="two" type=button value="预览图片">
            <input class="one" id="three" type=button value="图片地址">
            <input class="one" id="four" type=button value="清除" >
            </span>
            </td>
          </tr>
      </table>
    </td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，代码见关键技术。

秘笈心法

心法领悟 177：jQuery 框架中的 empty()方法。

empty()方法：删除匹配的元素集合中所有的子节点，语法格式如下：

```
$('class/id').empty();
```

参数说明：

class/id：表示类名或 ID 名称。

实例 178	去掉下拉列表框的边框 光盘位置：光盘\MR\03\178	高级 趣味指数：★★★★

实例说明

由于 CSS 样式对下拉列表框的定义很少，以至于想要修改下拉列表框的样式变得很复杂。本实例将讲解如何去掉下拉列表框的边框，如图 3.34 所示。

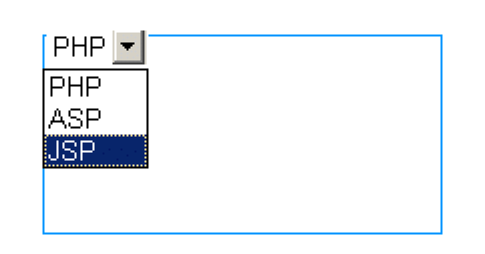


图 3.34　去掉下拉列表框的边框

关键技术

本实例的关键点是严格定义 CSS 样式，代码如下：

```
<div style="position: absolute; left: 10px; top: 10px; width: 115px; height: 20px;   clip:rect(2 114 20 2);">
```

设计过程

创建 index.php 文件，利用<table>标签编写表格边框。编辑 CSS 样式，通过 CSS 样式去掉下拉列表框的边框，代码如下：

```
<table width="200px" height="100px" border="1" bordercolor="#0099FF" cellspacing="0"><tr><td>
<div style="position: absolute; left: 10px; top: 10px; width: 115px; height: 20px;   clip:rect(2 114 20 2);">
    <select>
      <option >PHP</option>
      <option >ASP</option>
    <option >JSP</option>
   </select>
</div>
</td></tr></table>
```

秘笈心法

心法领悟 178：CSS 的效果。

像上面这样较为生僻的代码，其实笔者也不能够完全“吃透”，但是没关系，因为我们所从事的是后台的开发，前台的东西了解一下即可。

实例 179	修改表单属性为弹出对话框 光盘位置：光盘\MR\03\179	高级 趣味指数：★★★★

实例说明

表单可以添加属性并且与客户端脚本 JavaScript 交互，如 JavaScript 的 click、change 事件等。运行本实例，当单击文本框时弹出提示信息，如图 3.35 所示。



图 3.35　弹出对话框

关键技术

本实例主要是通过 JavaScript 代码定义单击事件，核心代码如下：

```
$(document).ready(function(){
$("#o").click(function(){
        alert("文本框被单击");
```



```
});
});
```

设计过程

创建 index.php 文件，引入 CSS 样式、JavaScript 脚本文件 in.js 和 jQuery 库文件 jquery-1.3.2.js。创建 form 表单并定义事件，代码如下：

```
<table align="center">
  <tr>
    <td>用户名：</td>
    <td><input id="o" type="text" /></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td><input id="o" type="text" onclick="alert('密码框被单击')"/></td>
  </tr>
</table>
```

秘笈心法

心法领悟 179：定义 JavaScript 事件。

如果用户想利用 JavaScript 本身的方法定义事件，需要在表单中进行，代码格式如实例 180 所示。如果是利用 jQuery 框架定义事件，只需要在标签中定义 ID 或者类即可。

```
<input id="o"type="text" onclick="alert('密码框被单击')"/>
```

实例 180	表单输入单元的文字设置 光盘位置：光盘\MR\03\180	高级 趣味指数：★★★★

实例说明

灵巧的 CSS 是所有 Web 开发者追捧的对象，其主要作用是通过简单的代码美化页面，页面美化用 CSS 基本都可以实现。本实例主要讲解通过定义 CSS 样式对输入单元的文字进行设置，如图 3.36 所示。

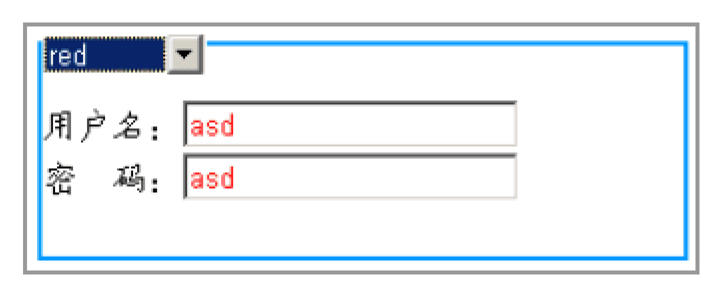


图 3.36　动态改变文字颜色

关键技术

本实例的关键点是定义 CSS 样式，通过 JavaScript 代码动态地改变文本框的颜色。JavaScript 代码如下：

```
<script type="text/javascript" language="javascript">
$(document).ready(function(){
        $("select").change(function(){
                var value = $("select").val();
                if(value == ""){
                        alert("请选择样式颜色");
                }else{
                        $("input:lt(2)").css("color",value);
                }
        });
});
</script>
```

设计过程

（1）创建 index.php 文件，利用 CSS 样式定义一个无边框的下拉列表框，代码如下：

```
<div style="position: absolute; left: 10px; top: 10px; width: 115px; height: 20px;    clip:rect(2 114 20 2);">
   <select>
     <option value="">样式颜色</option>
     <option value="red">red</option>
     <option value="blue">blue</option>
     <option value="yellow">yellow</option>
   </select>
</div>
```

（2）编写文本框和密码框，代码如下：

```
<div style="font-family:'楷体_GB2312'; font-size:16px;">
用户名：<input type="text" /><br>
密  码：<input type="text" />
</div>
```

（3）编写 JavaScript 脚本文件 in.js，动态改变文本框输入单元颜色，代码见关键技术。

秘笈心法

心法领悟 180：jQuery 中匹配所有小于给定索引值的元素。

例如，$("tr:lt(2)")表示匹配<tr>标签中小于索引值 2 的元素。

实例 181	表单输入单元单击删除 光盘位置：光盘\MR\03\181	高级 趣味指数：★★★★

实例说明

在创建表单中的文本框等内容时，往往会给文本框设定初始值，这些初始值需要用户自行手动删除。为了方便用户，节省操作时间，可以通过设置 JavaScript 事件，实现直接删除文本框中初始值的功能。运行本实例，效果如图 3.37 所示。



图 3.37　表单输入单元单击删除

关键技术

本实例主要是通过 JavaScript 的 click 事件实现，当单击文本框时删除文本框中的默认值，代码如下：

```
$(document).ready(function(){
 $("input").click(function(){
        $(this).attr("value","");
```



```
        alert('清除文本框信息');
});
$("textarea").click(function(){
        $(this).attr("value","");
});
});
```

设计过程

创建 index.php 文件，引入 CSS 样式、JavaScript 脚本文件 in.js 和 jQuery 库文件 jquery-1.3.2.js。编写表单信息，代码如下：

```
<table align="center">
  <tr>
    <td>标题：</td>
    <td><input type="text" value="请输入标题" /></td>
  </tr>
  <tr>
    <td>内容：</td>
    <td><textarea cols="23" rows="3">请输入内容</textarea></td>
  </tr>
</table>
```

秘笈心法

心法领悟 181：如何简单实现本实例功能。

本实例是采用 jQuery 框架实现的，书写代码较多，还有一种简便方法是在表单中直接定义 onFocus 事件，代码如下：

```
<input type='text' name='te' value='请输入内容' onFocus="this.value=''">
```

实例 182	表单文本输入的移动选择 光盘位置：光盘\MR\03\182	高级 趣味指数：★★★★

实例说明

很多网站中往往有一些小的技巧，例如，在文本域中输入内容时，通过下拉列表框动态地将文字插入文本域中。下面向用户讲述这一功能。运行本实例，效果如图 3.38 所示。

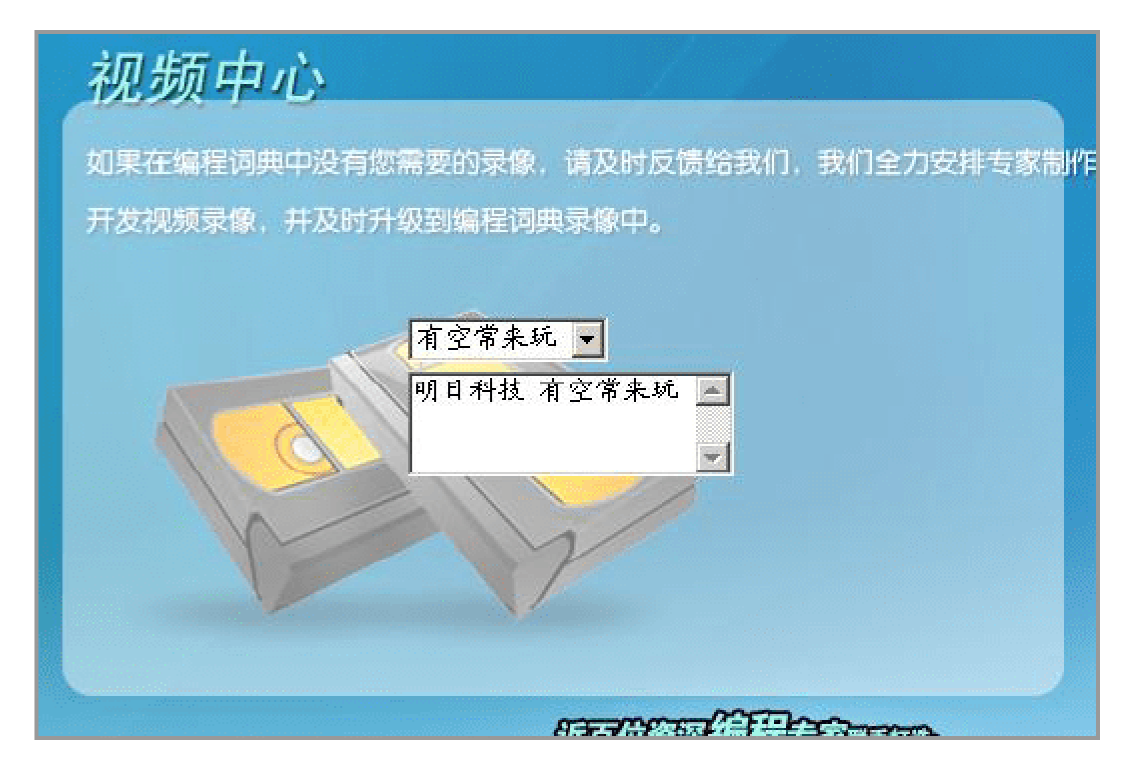


图 3.38　表单文本输入的移动选择

关键技术

本实例通过 JavaScript 脚本中的 change 事件，实现动态插入信息，其核心代码如下：

```
$(document).ready(function(){
$("textarea").select();                          //选择下拉列表框
$("select").change(function(){                   //定义事件
        var value = $(this).val();               //获取 value 值
        var va = $("textarea").val();            //获取文本域的值
        if(va == "请输入内容"){
                $("textarea").html(value);       //输出文本域的值
        }else{
                $("textarea").html(va+value);    //为文本域赋值
        }
});
});
```

设计过程

（1）创建 index.php 文件，引入 CSS 样式和 JavaScript 脚本文件 in.js，以及 jQuery 库，创建下拉列表框和文本域，代码如下：

```
<table align="center">
  <tr>
    <td><select name="select">
      <option value="谢谢">谢谢</option>
      <option value="不客气">不客气</option>
      <option value="有空常来玩">有空常来玩</option>
      <option value="再见">再见</option>
</select></td>
  </tr>
  <tr>
    <td><textarea cols="20" rows="3">请输入内容</textarea></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，代码见关键技术。

秘笈心法

心法领悟 182：如何利用 jQuery 实现文本框内容全选。

实现文本框内容全选与单击文本框清除所有信息所达到的效果大同小异，其实现代码如下：

```
$("textarea").select();
```

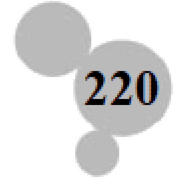
实例 183	通过下拉列表框选择头像 光盘位置：光盘\MR\03\183	高级 趣味指数：★★★★

实例说明

在腾讯的 QQ 系统中，可以通过单击头像实现更换新头像的功能。其实这个功能相对来说是比较简单的。运行本实例，通过下拉列表框实现头像的选择，如图 3.39 所示。



图 3.39　选择头像



关键技术

本实例通过 JavaScript 的 change 事件和实例 162 中的 2 级联动菜单实现，其核心代码如下：

```
$(document).ready(function(){
$("select").change(function(){
        var value = $(this).val();
        if(value == ""){
                alert("请选择性别");
        }else{
                if(value == "nan"){
                        $("#o").empty();
                        $("#o").append("<select id='one'><option value='pic/1.jpg'>头像 1</option><option value='pic/2.jpg'>头像 2</option><option value='pic/3.jpg'>头像 3</option><option value='pic/4.jpg'>头像 4</option><option value='pic/9.jpg'>头像 5</option></select>");
                        $("#n").empty();
                        $("#n").append("<img src='pic/1.jpg'>");
                        $("#one").change(function(){
                                var va = $(this).val();
                                $("#n").empty();
                                $("#n").append("<img src="+va+">");
                        });
                }else{
                        $("#o").empty();
                        $("#o").append("<select id='one'><option value='pic/5.jpg'>头像 1</option><option value='pic/6.jpg'>头像 2</option><option value='pic/7.jpg'>头像 3</option><option value='pic/8.jpg'>头像 4</option><option value='pic/10.jpg'>头像 5</option></select>");
                        $("#n").empty();
                        $("#n").append("<img src='pic/5.jpg'>");
                        $("#one").change(function(){
                                var va = $(this).val();
                                $("#n").empty();
                                $("#n").append("<img src="+va+">");
                        });
                }
        }
});
});
```

设计过程

（1）创建 index.php 文件，引入 CSS 样式和 JavaScript 脚本文件 in.js，以及 jQuery 库，编写下拉列表和标签，代码如下：

```
<table align="center">
  <tr>
    <td>
      <select name="select">
            <option value="">请选择性别</option>
            <option value="nan">男性</option>
            <option value="nv">女性</option>
      </select>
      <div id="o"></div><div id="n"></div>
</td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，代码见关键技术。

秘笈心法

心法领悟 183：jQuery 中的 this 关键字。

jQuery 中的 this 关键字与其他编程语言大致相同，此关键字表示当前方法体的类名称或者 ID 和标签，书写格式如下：

```
var va = $(this).val();
```

3.5　CSS+DIV 页面布局

CSS+DIV 是一种网站的标准（或称面向 Web 的标准）中常用术语之一，通常是为了说明与 HTML 页面中的表格（table）定位技术的区别，或者是因为 HTML 网站设计标准中不能使用表格定位技术，而采用 CSS+DIV 方式实现的各种定位。

实例 184	应用 CSS+DIV 实现无边框窗口 光盘位置：光盘\MR\03\184	高级 趣味指数：★★★★

实例说明

对于一个页面风格清新自然的网站来说，如果打开的窗口带有不适合页面风格的灰色边框及死板的标题栏，不但不能体现其个性化，而且势必会影响网站的整体效果。本实例介绍一种应用 CSS+DIV 实现的无边框窗口。运行本实例，单击“用户登录”超链接，即可打开无边框的用户登录窗口，如图 3.40 所示。

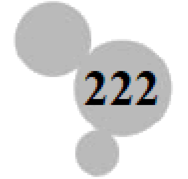


图 3.40　无边框窗口

关键技术

本实例通过 CSS+DIV 排版定位技术，将表格放置在指定位置，CSS 代码如下：

```
body{
 margin:0;
 text-align:center;
}
#container{
 position:relative;
 margin:10px 10px 10px 10px;
 width:600px;
 height:600px;
 text-align:center;
 background:url(../pic/bgc.jpg);
}
#NOborder{
 clear:left;
 float:right;
 position:absolute;
```



```
margin-right:10px;
width:200px;
height:100px;
margin:auto;
}
#border{
width:auto;
height:100px;
}
#href{
clear:left;
float:right;
position:absolute;
margin:40px 10px 300px 200px;
width:80px;
height:20px;
}
```

利用 JavaScript 代码实现表格的显示或隐藏，JavaScript 代码如下：

```
$(document).ready(function(){
$("a").click(function(){
        $("table").css("display","block");
});
$("#two").click(function(){
        $("table").css("display","none");
});
});
```

■ 设计过程

（1）创建 index.php 文件，载入 CSS 样式文件和 JavaScript 脚本文件，通过<div>标签对网页进行布局，编写无边框表格，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<div id="container">
<div id="border">
        <div id="href">
                <a href="#">用户登录</a>
        </div>
</div>
<div id="NOborder">
<table border="1" bordercolor="#999999" cellspacing="0"><tr><td>
<table>
   <tr>
      <td>==用户登录==</td>
   </tr>
   <tr>
      <td>用户名：<input type="text" size="10"/></td>
   </tr>
   <tr>
      <td>密  码：<input type="text" size="10" /></td>
   </tr>
   <tr>
      <td><input id="one" value="" type="button" /><input id="two" value="" type="button" /></td>
   </tr>
</table>
</td></tr></table>
</div>
</div>
```

（2）编写 CSS 样式文件 in.css 和 JavaScript 脚本文件 in.js，代码见关键技术。

■ 秘笈心法

心法领悟 184：CSS 中 position 关键字的使用。

position 关键字用于检索对象的定位方式，其属性值如下：

static：无特殊定位，对象遵循 HTML 定位规则。

absolute：将对象从文档流中拖出，使用 left、right、top、bottom 等属性进行绝对定位，而其层叠通过 z-index 属性定义。此时对象不具有边距，但仍有补白和边框。

relative：对象不可层叠，但将依据 left、right、top、bottom 等属性在正常文档流中偏移位置。

fixed：IE 5.5 及 NS 6 尚不支持此属性。

实例 185 设置表格样式

光盘位置：光盘\MR\03\185

高级

趣味指数：★★★★

实例说明

表格在 Web 开发中很常见，很多程序员都喜欢利用表格对自己数据库中的信息进行格式输出。但是表格在默认情况下，无论从色调还是格式上都不是很理想，这就需要程序员利用 CSS 的知识对其样式进行设置。运行本实例，效果如图 3.41 所示。

用户ID	用户名	密码	操作时间
001	ym	***	2010-08-12
002	pkh	***	2010-08-23
003	lzh	***	2010-07-06

图 3.41　设置表格样式

关键技术

本实例利用 CSS 样式，对表格的边框、背景色等进行设置，核心代码如下：

```
body{
 margin-top:150px;
}
#o{
 margin-left:150px;
}
table,select{
 font-family:"楷体_GB2312";
 font-size:16px;

}
table{
 background:#32ACEF;
}
td{
 text-align:center;
}
.one{
 background:#FFFFFF;
}
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式文件 in.css，利用<div>标签设计网页框架，代码如下：

```
<div id="o">
<div id="banner">
        <img src="pic/banner.jpg" />
</div>
<div id="e">
```



```
<table width="575" border="1" bordercolor="#7DD5FA" cellspacing="0"><tr><td>
<table width="575px"><tr>
      <td>用户 ID</td>
      <td>用户名</td>
      <td>密码</td>
      <td>操作时间</td>
  </tr>
  <tr class="one">
      <td>001</td>
      <td>ym</td>
      <td>***</td>
      <td>2010-08-12</td>
  </tr>
  <tr class="one">
      <td>002</td>
      <td>pkh</td>
      <td>***</td>
      <td>2010-08-23</td>
  </tr>
  <tr class="one">
      <td>003</td>
      <td>lzh</td>
      <td>***</td>
      <td>2010-07-06</td>
  </tr>
</table></td></tr></table>
```

（2）编写 CSS 样式，代码见关键技术。

秘笈心法

心法领悟 185：使用 CSS 中 margin 关键字的注意事项。

（1）margin 关键字主要用于检索和设置对象四边的外沿边距，此关键字根据传入参数的不同，性能也不相同。如果提供全部的 4 个参数值，将按上、右、下和左的顺序作用于四边；如果只提供 1 个参数，将用于全部的四边；如果提供 2 个参数，第 1 个用于上和下，第 2 个用于左和右；如果提供 3 个参数，第 1 个用于上，第 2 个用于左和右，第 3 个用于下。

（2）内联对象要使用该属性，必须先设定对象的 height 或 width 属性，或者设定 position 属性为 absolute。在 IE 4+中，margin 属性不可用于 td 和 tr 对象。

（3）外延边距始终透明。

实例 186	设置图片的样式 光盘位置：光盘\MR\03\186	高级 趣味指数：★★★★

实例说明

CSS 之所以受到很多 Web 设计者的追捧，是因为它几乎可以实现所有的基础美化功能，图片当然也不例外。运行本实例，应用 CSS 样式实现图片向左和向右浮动、为图片添加边框的功能，运行结果如图 3.42 所示。

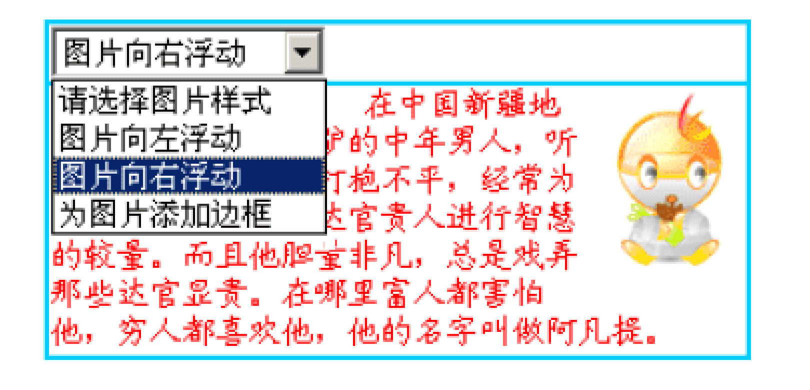


图 3.42　设置图片的样式

关键技术

本实例主要通过 JavaScript 代码动态地为图片添加 CSS 样式，核心代码如下：

```
$(document).ready(function(){
$("select").change(function(){
	var value = $(this).val();
	if(value == ""){
		alert("您没有选择图片样式");
	}else{
		if(value == 1){
			$("img").css("float","left");
		}else{
			if(value == 2){
				$("img").css("float","right");
			}else{
				$("img").css({border:"1px red solid"});
			}
		}
	}
});
});
```

设计过程

（1）创建脚本文件，利用<div>标签建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写下拉列表框和内容主体，代码如下：

```
<div>
<table border="1" bordercolor="#00CCFF" cellspacing="0"><tr><td>
<select>
	<option value="">请选择图片样式</option>
	<option value="1">图片向左浮动</option>
	<option value="2">图片向右浮动</option>
	<option value="3">为图片添加边框</option>
</select>
</td></tr><tr><td>
	<img src="pic/img.gif" />
	    很久很久以前，　在中国新疆地区，有一个骑着毛驴的中年男人，听说他智慧过人，爱打抱不平，经常为了老百姓的利益与达官贵人进行智慧的较量。而且他胆量非凡，总是戏弄那些达官显贵。在那里富人都害怕他，穷人都喜欢他，他的名字叫做阿凡提
</td></tr></table></div>
```

（2）编写 JavaScript 脚本，代码见关键技术。

秘笈心法

心法领悟 186：CSS 中如何使用 float 属性。

float 属性指定对象是否浮动和如何浮动等，参数如下：

☑ nne：对象不浮动。

☑ left：对象浮在左边。

☑ right：对象浮在右边。

实例 187	滚动条样式 光盘位置：光盘\MR\03\187	高级 趣味指数：★★★★

实例说明

用户在设计页面时，为了达到整体协调的效果，很多时候需要重新设计滚动条。本实例只是更改了滚动条相关的色调，没有对其整体效果进行更改，运行效果如图 3.43 所示。





图 3.43　滚动条样式

关键技术

本实例通过 CSS 样式对浏览器的滚动条等颜色进行更改，核心代码如下：

```
td{font-size:9pt}
BODY {
SCROLLBAR-BASE-COLOR: #FFAD00;
SCROLLBAR-FACE-COLOR: #FFCB29;
SCROLLBAR-ARROW-COLOR: #800529;
SCROLLBAR-TRACK-COLOR: #EEFFCC;
SCROLLBAR-DARKSHADOW-COLOR:#FFFFFF;
}
```

设计过程

（1）创建 index.php 文件，设置网页页面，载入 CSS 样式，其关键代码如下：

```
<link href="css/style.css" rel="stylesheet">
```

（2）编写 CSS 样式文件 style.css，代码见关键技术。

秘笈心法

心法领悟 187：CSS 中设置 color 属性的注意事项。

color 为要设置的颜色代码，可以是十六进制的，如#FF0000，也可以是以 RGB 表示的，如 RGB（255，0，255）。在设置滚动条样式时，不必把所有的属性都用上，用户可根据网页的需求自行设置。

实例 188	设置表单样式 光盘位置：光盘\MR\03\188	高级 趣味指数：★★★★

实例说明

表单几乎是所有网站都不可或缺的一个模块，用户可以用它登录到网站或者填写个人信息等。所以对于一个 Web 开发者来说，学习设置一个优秀的表单样式是很重要的。运行本实例，通过 CSS 设置表单样式，效果如图 3.44 所示。

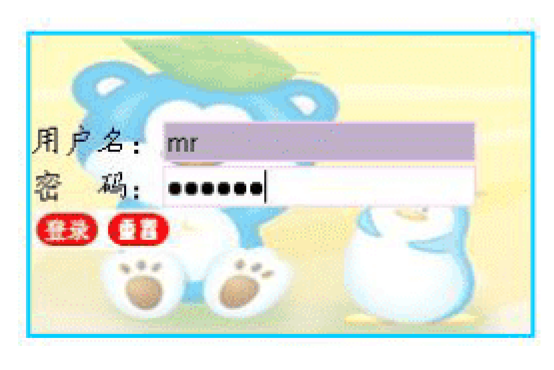


图 3.44　设置表单样式

关键技术

本实例通过 CSS 样式对表单的字体、标签、文本边框进行设置，实现美化的目的，代码如下：

```
body{
 margin:0px;
}
div{
 margin-top:150px;
}
#one{
 margin-left:400px;
 font-family:"楷体_GB2312";
 font-size:16px;
}
input{
 width:150px;
}
#e{
 width:35px;
 background-image:url(../pic/buttom.jpg);
 border:0;
}
#ee{
 width:35px;
 background-image:url(../pic/buttom1.jpg);
 border:0;
}
#o,#n{
 border:1px #FFCCFF solid;
}
table{
 width:250px;
 height:150px;
 background-image:url(../pic/bg.jpg);
}
```

另外还应用 JavaScript 特效，使鼠标覆盖或移出时文本框变色，代码如下：

```
$(document).ready(function(){
 $("input:lt(2)").mouseover(function(){
        $(this).css("background","#BFAFC9");
        $(this).mouseout(function(){
                $(this).css("background","white");
        });
 });
});
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式、JavaScript 脚本文件和 jQuery 类库文件，通过<div>定位技术对表格信息进行定位。创建 form 表单，设置用户名和密码文本框，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<div>
 <div id="one">
 <table border="1" bordercolor="#00CCFF" cellspacing="0"><tr><td>
 用户名：<input type="text" id="o" /><br>
 密  码：<input type="password" id="n" /><br>
 <input type="button" id="e" value=""/><input type="button" id="ee" value=""/>
 </td></tr></table>
 </div>
</div>
```

（2）编写 CSS 样式和 JavaScript 脚本，代码见关键技术。



秘笈心法

心法领悟 188：CSS 中设置 border 属性的注意事项。

border 属性是复合属性，如使用该复合属性定义单个参数，则其他参数的默认值将无条件覆盖各自对应的单个属性设置，默认值为 medium none。border-color 的默认值将采用文本颜色。要使用该属性，必须先设定对象的 height 或 width 属性，或者设定 position 属性为 absolute。

实例 189	设置超链接的样式 光盘位置：光盘\MR\03\189	高级 趣味指数：★★★★

实例说明

超链接在 Web 开发中也是很常见的，其主要作用是执行客户端的脚本跳转。默认情况下，单击之前颜色为蓝色，存在下划线样式；单击之后颜色为紫色，同样存在下划线样式。但是这种样式不可能适合所有的网站色调。本实例介绍如何更改超链接的样式，运行效果如图 3.45 所示。

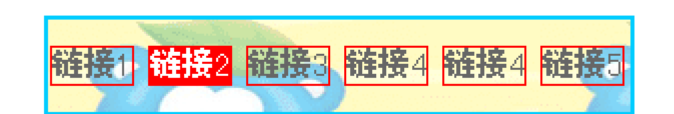


图 3.45　设置超链接的样式

关键技术

本实例通过 CSS 样式，定义<a>标签的 3 个属性，控制超链接的样式，其核心代码如下：

```
a{
 border:1px #FF0000 solid;
}
a:link{
 color:#333333;
 text-decoration:none;
}
a:visited{
 color:#666666;
 text-decoration:none;
}
a:hover{
 background-color:#FF0000;
 color:#FFFFFF;
}
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式文件，编写 5 个超链接，代码如下：

```
<div>
 <div id="one">
 <table border="1" bordercolor="#00CCFF" cellspacing="0" background="pic/bg.jpg"><tr><td>
 <a href="#">链接 1</a> <a href="#">链接 2</a> <a href="#">链接 3</a> <a href="#">链接 4</a> <a href="#">链接
4</a> <a href="#">链接 5</a>
 </td></tr></table>
 </div>
</div>
```

（2）编写 CSS 样式，代码见关键技术。

秘笈心法

心法领悟 189：定义<a>标签的 3 个属性。

☑ :link：设置 a 对象在未被访问前的样式表属性。
☑ :visited：设置 a 对象在其链接地址已被访问时的样式表属性。
☑ :hover：设置对象在鼠标悬停时的样式表属性。

实例 190 设置 body 的样式

光盘位置：光盘\MR\03\190

高级

趣味指数：★★★★

实例说明

HTML 将整个网页分成头部（head）和身体（body）两个部分，头部一般包含标题和引入的 JavaScript 和 CSS 样式等；身体部分主要是执行的代码，其中可以包含表格、表单和<div>标签等元素。对于<body>中的元素可以进行 CSS 的样式美化，但是如果用户想定义类似全文的一个文字的样式，则需要对每个标签里输出到页面的文字进行 CSS 样式的定义，这样未免太麻烦，因此可以在<body>标签中应用 CSS 样式，这样就可以对整个页面中的文字进行设置，运行结果如图 3.46 所示。



图 3.46　设置 body 的样式

关键技术

本实例通过定义<body>标签的 CSS 样式，对整个页面中的文字及文字的颜色进行定义，其核心代码如下：

```
body{
font-family:"楷体_GB2312";
font-size:14px;
color:#666666;
}
```

设计过程

（1）创建 index.php 文件，利用<table>标签建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table width="595px" height="448px" align="center" border="1" bordercolor="#0099FF" cellspacing="0" background="pic/bg.jpg">
  <tr>
    <td><table align="center">
  <tr>
    <td>用户名：<input type="text" id="o" /></td>
  </tr>
  <tr>
    <td>密  码：<input type="password"   id="o"/></td>
```



```
  </tr>
</table>
</td>
  </tr>
</table>
```

（2）编写 CSS 样式和 JavaScript 脚本文件，核心代码见关键技术。

秘笈心法

心法领悟 190：定义<body>标签属性的注意事项。

当用 CSS 样式定义了<body>标签后，如果想定义<body>内的元素属性，需要对每个元素进行更改。

实例 191	模拟的进度条 光盘位置：光盘\MR\03\191	高级 趣味指数：★★★★

实例说明

在一些大型网站的首页中，为了体现动画效果，通常在进入首页前添加一个模拟的进度条，通过该进度条实现打开网站首页的前奏。运行本实例，将自动加载一个模拟的进度条，运行结果如图 3.47 所示。当进度条完成 100%后，自动跳转到 http://www.mingrisoft.com 网站。



图 3.47　模拟的进度条

关键技术

本实例主要应用 for 循环语句控制进度条的进度，当进度条完成整个进度的显示时，自动调用 finish()函数，从而跳转到指定的网站。

另外，本实例中进度条的样式是通过 CSS 样式设置的，并应用 div 层输出到浏览器。

设计过程

（1）应用 CSS 样式设置 div 层及文字的样式，代码如下：

```
<style type="text/css">
div{font-size: 9pt}
#load{font-size: 9pt; cursor: default; position: absolute; display:block;width:402;height:20;top:expression((document.body.clientHeight-50)/2);
left:expression((screen.width-400)/2); display: block; z-index:100;background:#EDECE9}
.px1{border: 1px solid ;background-color: #FFFFFF;}
body {
 background-color: #000000;
}
.style1 {color: #FFFF00}
</style>
```

（2）在<body>中应用 div 层设计进度条，应用 class 为 div 层引用已定义的 CSS 样式，并添加部分说明文字，然后应用 for 循环语句限制进度条完成整个进度的行程，代码如下：

```
<body onload=finish()>
<div id=load style="background-color:#000000"><span class="style1">请稍候...</span>
<div class=px1><img   id=line style="background:red" width=14 height=14></div>
<div class="style1" id=load_txt>loading</div>
</div>
<?php
for($i=0;$i<500;$i++){
?>
```

```
<script>load_num("Loading");</script>
<?php
}
?>
</body>
```

（3）应用 JavaScript 脚本实现进度条的动态效果，当进度条完成整个进度的显示时，自动调用 finish()函数，从而跳转到指定的网站，代码如下：

```
<script language=JavaScript>
var load_line_i=1;
var load_line_step=500;
function load_num(txt)
{load_line_i+=400/load_line_step;
load_txt.innerText=txt+" "+Math.floor(load_line_i/4.00)+"%";
document.all("line").width=load_line_i;}
function finish(){
window.location.href="http://www.mingrisoft.com";
}
</script>
```

秘笈心法

心法领悟 191：JavaScript 脚本客户端跳转的两种形式。

当页跳转：

```
window.location.href="http:// "
```

新页跳转：

```
Window.open="http://"
```

实例 192	<div>标签设计用户注册页面 光盘位置：光盘\MR\03\192	高级 趣味指数：★★★★

实例说明

在进行网页排版时，用户可以使用<table>标签进行布局，但是在比较大的网页中，用<table>标签布局会使页面代码变得混乱，不容易解读，所以笔者建议尽量使用<div>标签进行页面布局。运行本实例，效果如图 3.48 所示。



图 3.48　<div>标签设计用户注册页面

关键技术

本实例的关键点是对<div>标签定位的灵活运用，及运用 CSS 样式控制页面的样式，核心代码如下：

```
body{
margin:0px;
text-align:center;
}
```



```
.one{
margin-top:100px;
width:595px;
height:448px;
background:url(../pic/bg.jpg);
}
input{
width:150px;
height:18px;
border-left:0;
border-top:0;
border-right:0;
border-bottom:1px #CCCCCC solid;
background:#6BACCC;
}
.two{
margin-top:200px;
margin-left:200px;
font-family:"楷体_GB2312";
font-size:16px;
color:#0000FF;
}
#pic{
border:0;
background:url(../pic/bt.jpg);
width:39px;
}
.three{
margin-right:170px;
}
```

设计过程

（1）创建脚本文件，利用<div>标签对网页进行排版，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下：

```
<div class="one">
<div class="two">
        <div>
                昵称：<input type="text" />
        </div>
        <div>
                密码：<input type="password" />
        </div>
        <div>
                QQ：<input type="text" />
        </div>
        <div>
                住址：<input type="text" />
        </div>
        <div class="three">
                <input id="pic" type="submit" value="" />
        </div>
</div>
</div>
```

（2）编写 CSS 样式，设置文本标签和标签位置定位，代码见关键技术。

秘笈心法

心法领悟 192：如何定义文本框只显示为一条下划线的 CSS 样式风格。

使文本框只显示为一条下划线的操作很简单，其原理是将文本框的左、右、上边框的 border 属性都设置为 0，然后定义下边框的 CSS 样式。以本实例为例，代码如下：

```
border-left:0;
border-top:0;
border-right:0;
border-bottom:1px #CCCCCC solid;
```

实例 193 <div>标签设计论坛帖子浏览页面

光盘位置：光盘\MR\03\193

高级

趣味指数：★★★★

实例说明

论坛的浏览页面相信对于所有用户来说并不陌生，其主要功能是将用户发表的内容供所有人或部分人浏览。本实例讲解的主要内容并不是如何实现浏览这一功能，而是对浏览页面的<div>布局。运行本实例，效果如图 3.49 所示。

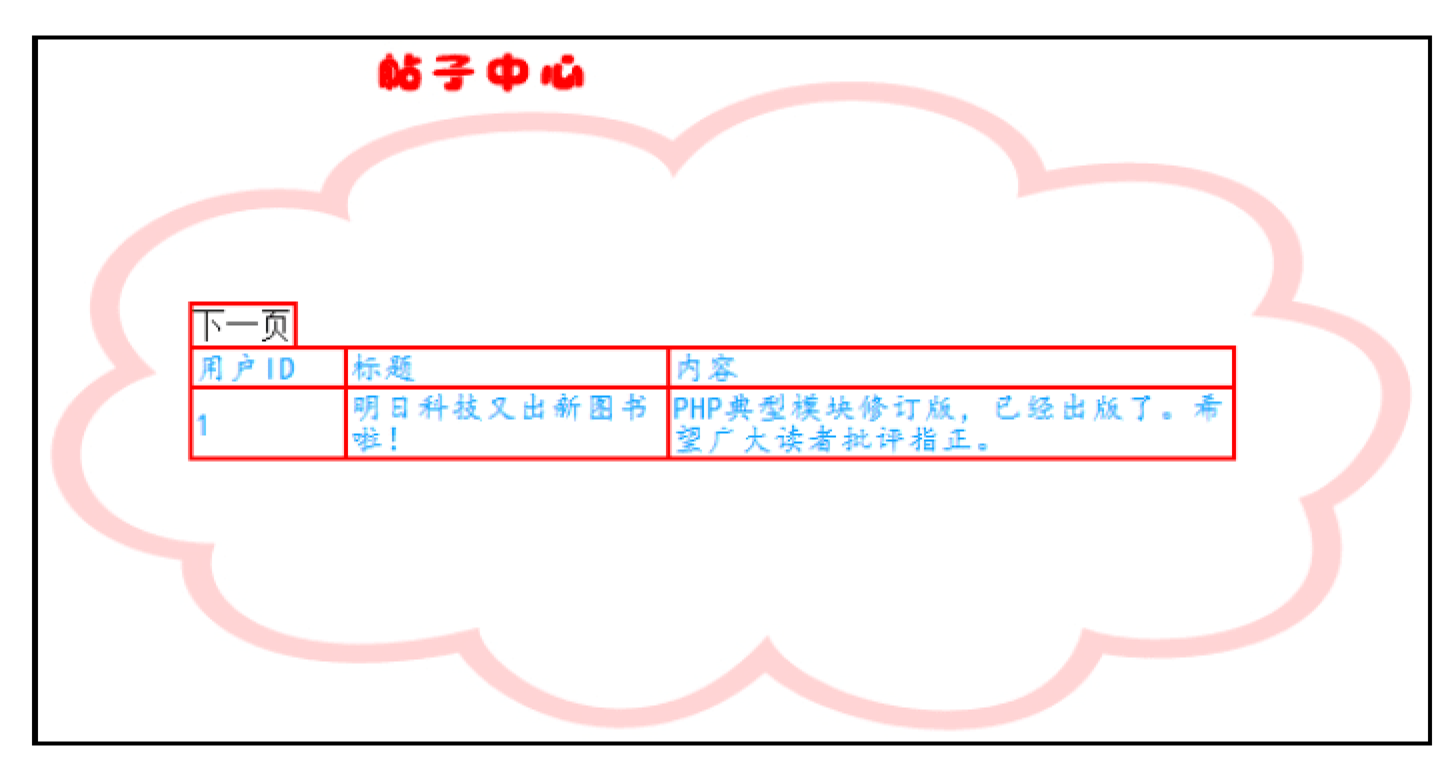


图 3.49 论坛帖子的浏览页面

关键技术

本实例的关键点是利用<div>的定位技术对页面进行合理布局，其中实现定位的 div 元素控制的代码如下：

```
.one{
margin:auto;
width:700px;
height:500px;
background:url(../pic/bg.jpg);
}
.three{
margin-top:200px;
margin-left:100px;
}
.five{
margin-left:100px;
}
```

设计过程

（1）创建脚本文件，引入 CSS 样式，连接数据库，完成数据库中数据的分页输出，代码如下：

```
<div class="one">
<div class="two">
<?php
$conn = mysql_connect("localhost","root","111") or die ("Connect MySQl False");        //连接数据库
mysql_select_db("db_database03",$conn) or die ("Connect db False");
mysql_query("SET NAMES UTF8");
$num = 1;
$rs = mysql_query("select * from tb_demo01");
$mu = mysql_num_rows($rs);
if(isset($_GET[page])){
        $page = $_GET[page];
```



```
}else{
        $page = 1;
}
if($page < $mu){                                                    //分页
?>
<div class="three">
<a href="index.php?page=<?php echo ($page+1);?>">下一页</a>
</div>
<?php
}else{
?>
<div class="three">
<a href="index.php?page=<?php echo ($page-1);?>">上一页</a>
</div>
<?php
}
$pag = $page - 1;
$sqlstr = "select * from tb_demo01 limit $pag,1";
$rss = mysql_query($sqlstr);
$rey = mysql_fetch_array($rss);
?>
<div class="five">
<table width="500px" border="1" bordercolor="#FF0000" cellspacing="0">      //数据输出
  <tr>
    <td width="70px">用户 ID</td><td width="150px">标题</td><td>内容</td>
  </tr>
  <tr>
    <td><?php echo $rey[id]?></td><td><?php echo $rey[title]?></td><td><?php echo $rey[content]?></td>
  </tr>
</table>
</div>
</div>
```

（2）编写 CSS 样式，代码如下：

```
body{
margin:0px;
}
.one{
margin:auto;
width:700px;
height:500px;
background:url(../pic/bg.jpg);
}
.three{
margin-top:200px;
margin-left:100px;
}
td{
font-family:"楷体_GB2312";
font-size:15px;
color:#0099FF;
}
.five{
margin-left:100px;
}
a{
font-family:"楷体_GB2312";
border-bottom:0;
border-left:2px #FF0000 solid;
border-right:2px #FF0000 solid;
border-top:2px #FF0000 solid;
}
a:link{
text-decoration:none;
color:#000000;
}
```

```
a:visited{
text-decoration:none;
color:#000000;
}
a:hover{
background-color:#FF0000;
color:#FFFFFF;
}
```

秘笈心法

心法领悟 193：相对路径与绝对路径。

相对路径与绝对路径是在日常程序开发过程中经常用到的两个概念。相对路径是指以当前文件为参照物，其他文件的位置；绝对路径指文件的绝对位置，即文件在本机中的绝对位置。以本实例为例，以 background: url(../pic/bg.jpg);引入背景图片，此路径为相对路径。

实例 194	<div>标签设计论坛后台管理页面 光盘位置：光盘\MR\03\194	高级 趣味指数：★★★★

实例说明

所有的 Web 开发者几乎都是从最简单的 HTML 语言开始学起，所以较<div>标签来说，程序员更喜欢用<table>标签来完成网页框架的布局。但是从网站优化的角度来说，<div>标签确实优于<table>标签。本实例通过设计后台管理页面，着重阐述如何使用<div>布局。运行本实例，效果如图 3.50 所示。

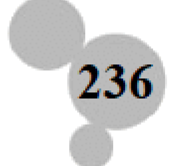


图 3.50　后台管理页面

关键技术

本实例利用<div>标签进行网页布局，并利用 CSS 定义宽、高等属性。实现定位的核心代码如下：

```
body{
margin:0px;
font-family:"华文彩云";
font-size:16px;
color:#000000;
}
.a{
margin:auto;
width:1023px;
```



```
height:600px;

}
.b{
margin:0px;
width:1023px;
height:80px;
background:url(../pic/top.jpg);
}
.c{
margin:0px;
width:165px;
height:310px;
position:absolute;
background:url(../pic/left.jpg);

}
.d{
margin:0;
width:845px;
height:300px;
background:url(../pic/right.JPG);
float:right;
}
.e{
background:url(../pic/down.JPG);
width:1023px;
height:10px;
}
```

设计过程

（1）创建 index.php 文件，利用<div>标签对网页进行布局，代码如下：

```
<div class="a">
<div class="b"></div>
<div class="c">        </div>
<div class="d">
</div>
<div class="e"></div>
</div>
```

（2）编写 CSS 样式，代码见关键技术。

秘笈心法

心法领悟 194：CSS 样式中，如何使文本或标签居中显示。

在<div>标签定位中，如果想使<div>标签内部的内容或标签居中显示，可以使用 CSS 中的 text-align 属性实现。

实例 195	<div>标签设计页面布局 光盘位置：光盘\MR\03\195	高级 趣味指数：★★★★

实例说明

在设计网页的过程中，能否控制好各个模块的位置是很重要的，而控制这些模块的标签中<div>是最常用的标记。利用<div>标签再加上 CSS 对其进行样式控制，可以实现很多效果。下面讲解<div>标签的使用特点和应用技巧。运行本实例，效果如图 3.51 所示。

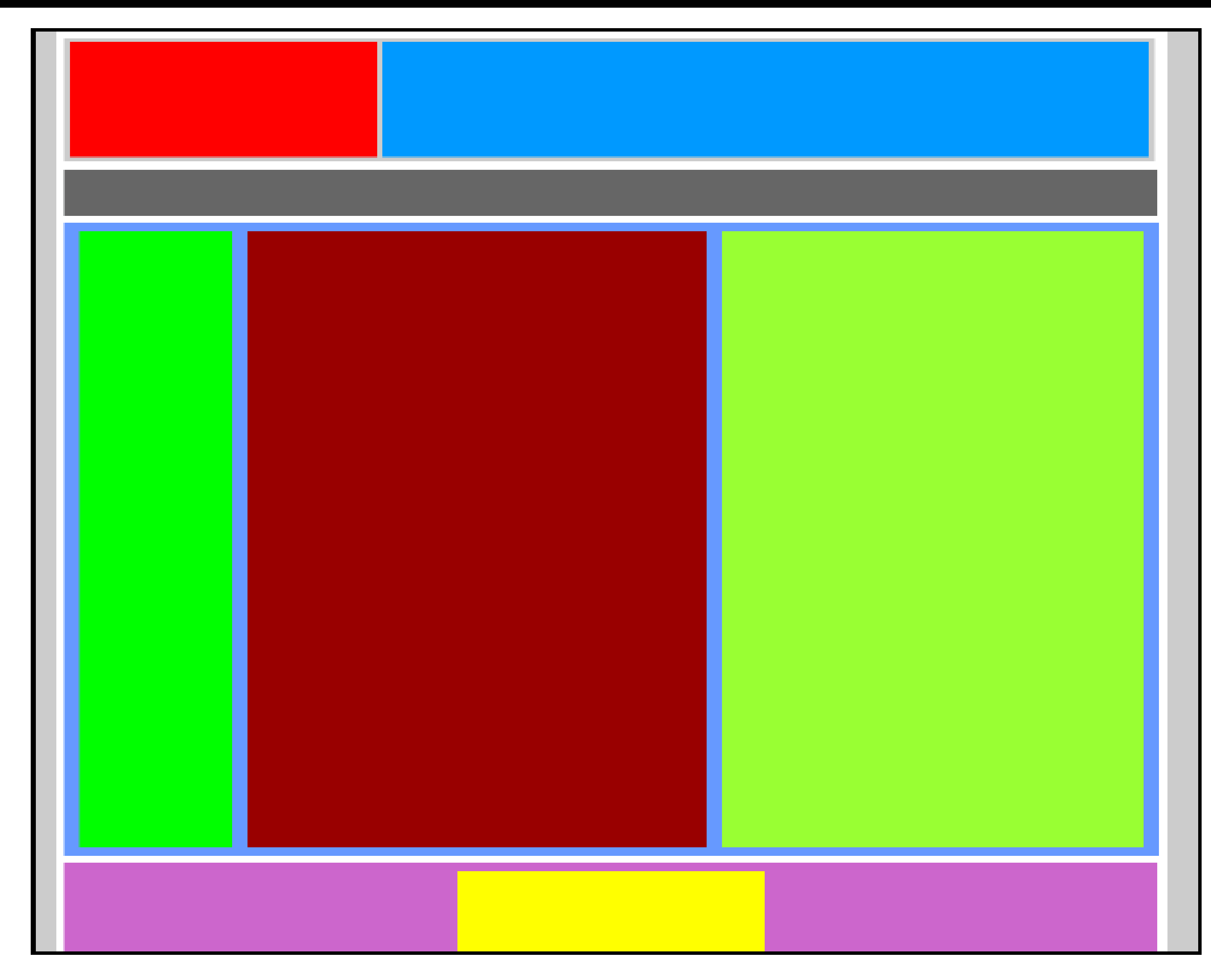


图 3.51　利用<div>标签布局

关键技术

本实例用到的属性包括 margin、width、height、float、clear，具体说明如下：

（1）margin：控制模块与外边距的距离。其中，如果只定义 1 个参数，表示到四边的边距；定义 3 个参数表示到上、左/右、下的边距；定义 2 个参数表示到上/下和左/右的边距。

（2）width：定义<div>模块的宽度。

（3）hight：定义<div>模块的高度。

（4）float：指定<div>对象模块如何浮动，其中包括 3 个参数，分别是 left、right、none，表示向左浮动、向右浮动、不浮动。

（5）clear：指定是否允许浮动对象，其中包括 4 个参数，即 left、right、both、none，分别表示清除左侧浮动、清除右侧浮动、清除两侧浮动、不清除浮动。

设计过程

（1）创建脚本文件，利用<div>定位技术设计网页结构，代码如下：

```
<div class="a">
	<div class="b">
		<div class="f"></div>
		<div class="j"></div>
	</div>
	<div class="c"></div>
	<div class="d">
		<div class="g"></div>
		<div class="h"></div>
		<div class="i"></div>
	</div>
	<div class="e">
		<div class="k"></div>
	</div>
</div>
```

（2）编写 CSS 样式，将<div>模块定位到指定位置并添加背景颜色，代码如下：

```
body{
 margin:0px;
 background:#CCCCCC;
```



```
}
.a{
margin:auto;
width:700px;
height:600px;
background:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:400px;
background:#6699FF;
}
.g{
margin:5px;
width:100px;
height:400px;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:300px;
height:400px;
background:#990000;
float:left;
}
.i{
margin:5px;
width:275px;
height:400px;
background:#99FF33;
float:right;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
```

```
width:200px;
height:90px;
background:#FFFF00;
}
```

秘笈心法

心法领悟 195：clear 属性的具体应用。

假设外部<div>标签定义宽度为 400 个像素，内部定义的两个<div>标签的总宽度为 450 个像素，如果内部右侧的<div>标签不能清除左侧的浮动的话，将会产生换行效果。

实例 196　页面布局（2 列左窄右宽高度自适应）

光盘位置：光盘\MR\03\196

高级

趣味指数：★★★★

实例说明

在比较小的博客中，网页一般分为上、中、下 3 个部分，中间部分又分左、右两个部分。如果利用 HTML 的<table>标签实现，则需要多个<table>标签进行嵌套，但是如果利用<div>标签，结构就比较明朗了。下面来实现这一实例，运行结果如图 3.52 所示。

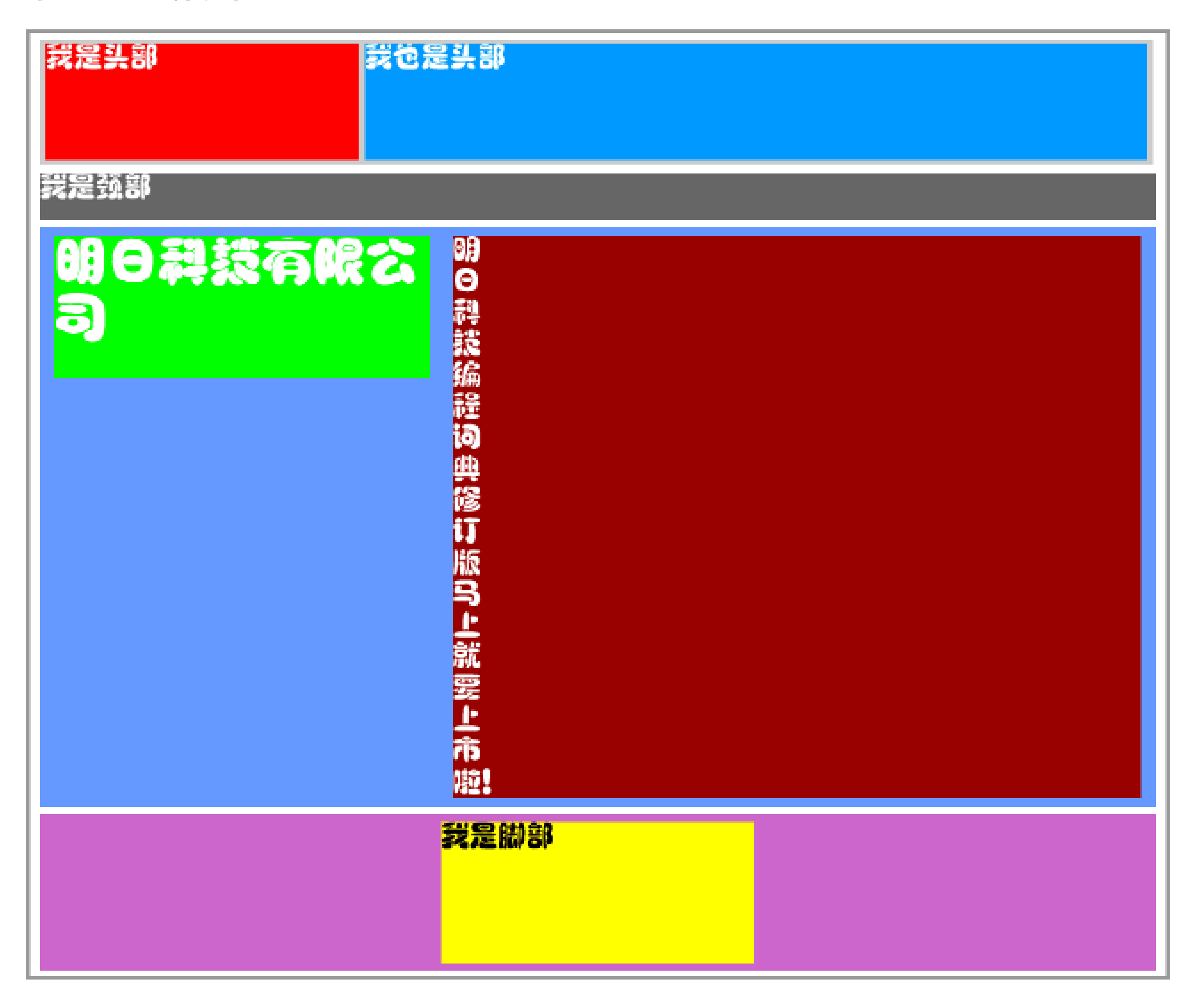


图 3.52　页面布局之 2 列左窄右宽高度自适应

关键技术

本实例的关键点是如何实现<div>模块的高度自适应。其实实现这一功能很简单，只需要定义 height 属性为 auto 即可，代码如下：

```
height:auto;
```

设计过程

（1）创建脚本文件，利用<div>标签对网页进行布局并输出提示文字，代码如下：

```
<div class="a">
        <div class="b">
```



```
                <div class="f"></div>
                <div class="j"></div>
        </div>
        <div class="c"></div>
        <div class="d">
                <div class="g">
                        明日科技<br>PHP 开发实录<br>典型模块大全<br>
                </div>
                <div class="h">
                        明<br>日<br>科<br>技<br>编<br>程<br>词<br>典<br>修<br>订<br>版<br>马<br>上<br>就<br>要<br>上<br>市<br>啦！
                </div>
        </div>
        <div class="e">
                <div class="k"></div>
        </div>
</div>
```

（2）编写 CSS 样式，定义<div>模块的宽度和高度，并为输出的文字定义样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:600px;
background:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:400px;
background:#6699FF;
}
.g{
margin:5px;
width:200px;
height:auto;
background:#00FF00;
float:left;
font-family:"楷体_GB2312";
font-size:16px;
}
```

```
.h{
margin:5px;
width:485px;
height:auto;
background:#990000;
float:right;
clear:left;
font-family:"楷体_GB2312";
color:#FFFFFF;
font-size:18px;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
}
```

秘笈心法

心法领悟 196：什么是高度自适应。

所谓的高度自适应是指根据文本内容的多少自动调整相应的高度，这样设置的优点是可以完全不浪费网页空间，而且使页面效果饱满丰富。高度自适应在网页设计中被广泛使用。

实例 197　页面布局（2 列固定宽度+头部+导航+尾部）

光盘位置：光盘\MR\03\197

高级

趣味指数：★★★★

实例说明

实例 196 已经讲解了页面布局中左窄右宽的情况，本实例介绍 2 列固定宽度的实现方法。运行本实例，效果如图 3.53 所示。

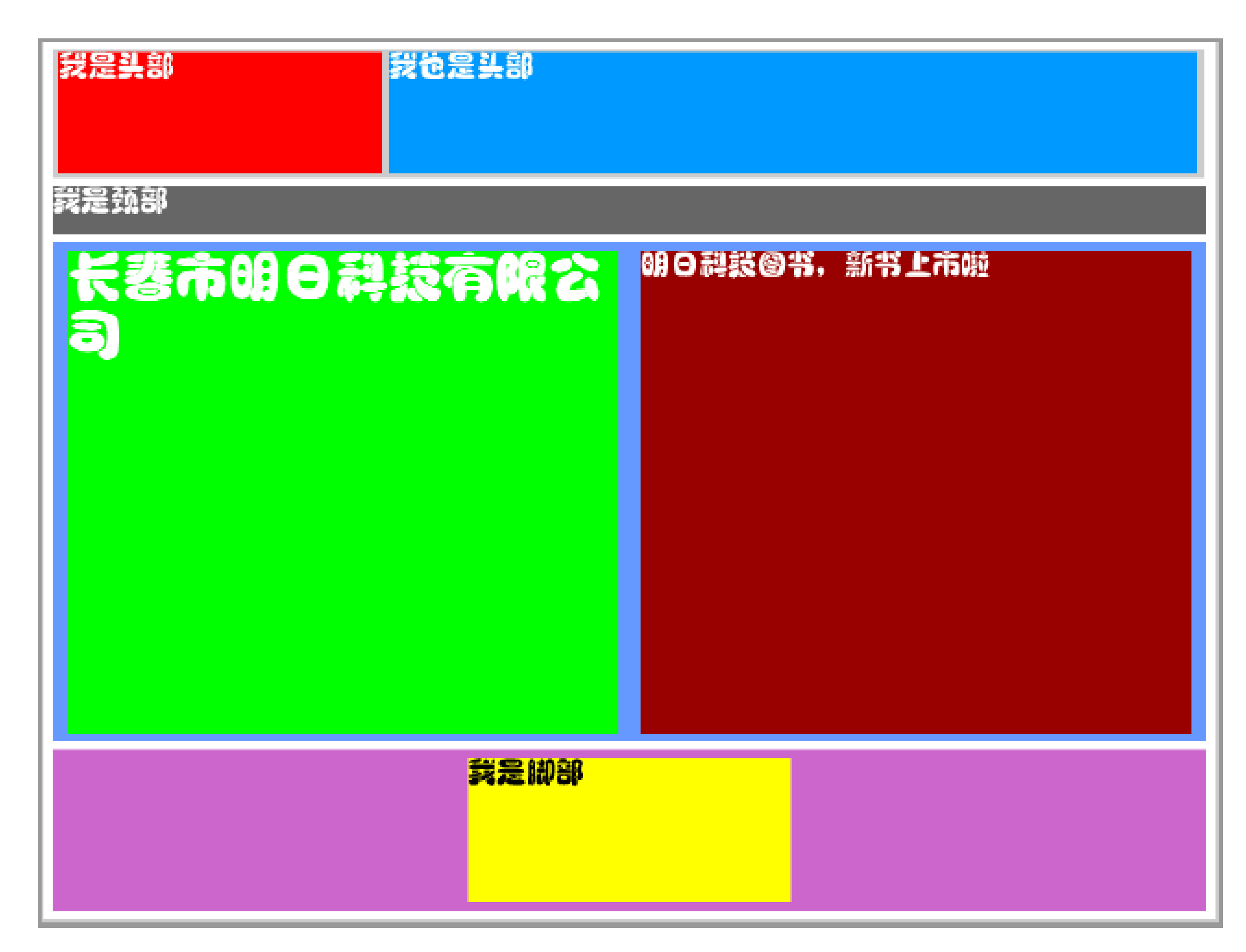


图 3.53　页面布局之 2 列固定宽度+头部+导航+尾部



关键技术

本实例设定固定宽度的方法是定义 width 属性为固定值，代码如下：

```
.g{
margin:5px;
width:340px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:340px;
height:300px;
background:#990000;
float:right;
clear:left;
}
```

设计过程

（1）创建脚本文件，利用<div>标签对网页进行布局并输出提示文字，代码如下：

```
<div class="a">
        <div class="b">
                <div class="f">我是头部</div>
                <div class="j">我也是头部</div>
        </div>
        <div class="c">我是颈部</div>
        <div class="d">
                <div class="g"><h1>长春市明日科技有限公司</h1></div>
                <div class="h">明日科技图书，新书上市啦</div>
        </div>
        <div class="e">
                <div class="k">我是脚部</div>
        </div>
</div>
```

（2）编写 CSS 样式，定义<div>相关属性并定义页面输出字体的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
```

```
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:340px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:340px;
height:300px;
background:#990000;
float:right;
clear:left;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000;
}
```

秘笈心法

心法领悟 197：float 属性的使用方法。

在定义身体部分时，如果身体部分只有两部分，可以利用 float 属性，一侧向左浮动，一侧向右浮动，但是如果身体部分分为 3 个部分，那么定义为以左面起，前两个向左浮动，最后一个向右浮动为好。

实例 198	页面布局（3 列左右侧固定中间自适应居中） 光盘位置：光盘\MR\03\198	高级 趣味指数：★★★★

实例说明

实例 196 和实例 197 都是以中间为两个模块进行布局。本实例讲解中间为 3 个模块的实例，并且实现左右侧固定，中间自适应居中显示，运行效果如图 3.54 所示。



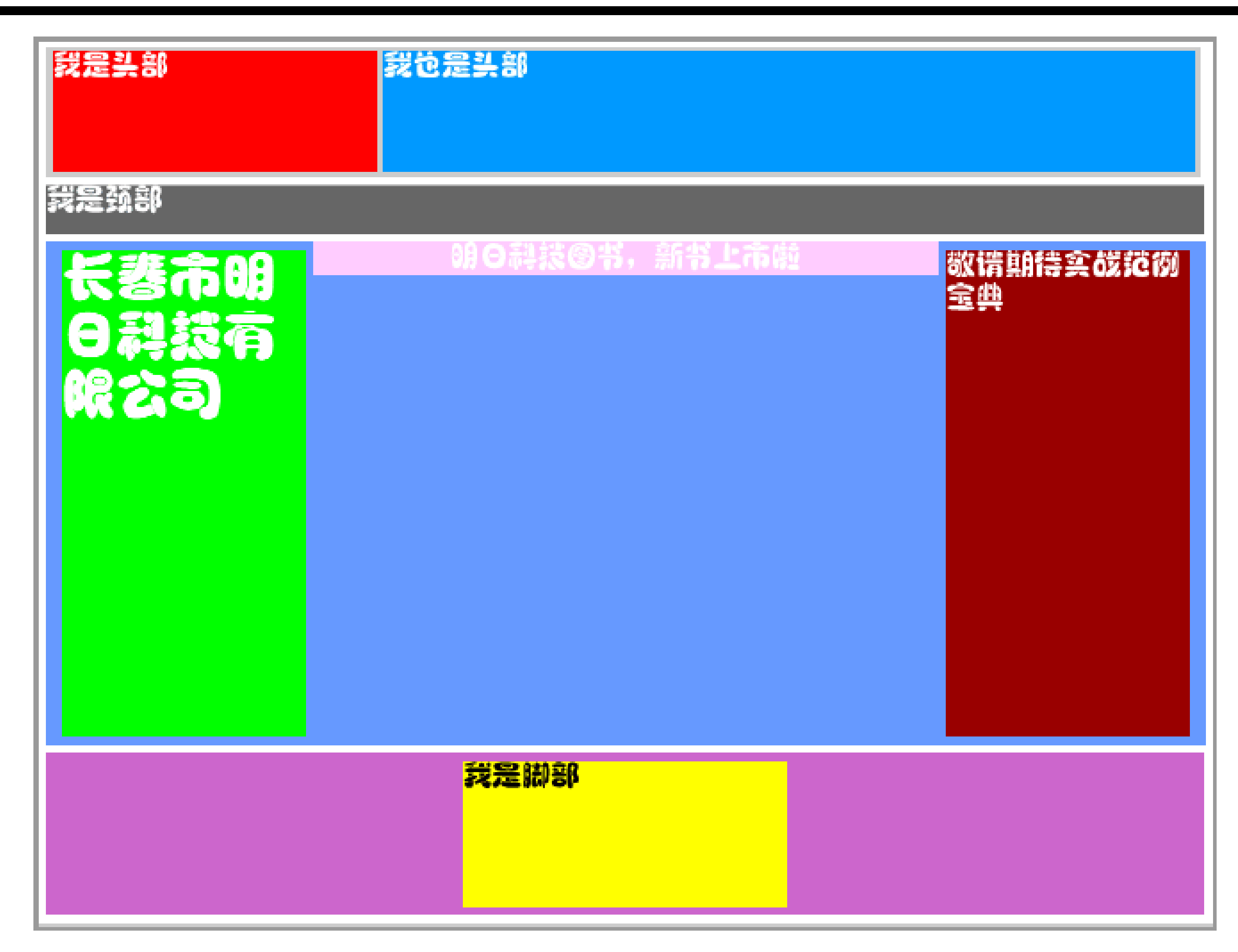


图 3.54　页面布局之 3 列左右侧固定中间自适应居中

关键技术

本实例的关键点是如何实现中间模块自适应居中。答案很简单，将模块的 margin 属性设置为上下方向与边距为 5 个像素，左右方向为 auto 自动设置即可，核心代码如下：

```
.h{
margin:5px auto;
width:385px;
height:auto;
background:#FFCCFF;
float:left;
text-align:center;
}
```

设计过程

（1）创建脚本文件，利用<div>标签进行网页定位布局，其中包括头部、颈部、身体部分和脚部，代码如下：

```
<div class="a">
<div class="b">
        <div class="f">我是头部</div>
        <div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
        <div class="g">
                <h1>长春市明日科技有限公司</h1>
        </div>
        <div class="h">
                明日科技图书，新书上市啦
        </div>
        <div class="i">
                敬请期待实战范例宝典
        </div>
</div>
<div class="e">
        <div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的宽和高，并定义页面输出字体的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:150px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px auto;
width:385px;
height:auto;
background:#FFCCFF;
float:left;
text-align:center;
}
.i{
margin:5px;
width:150px;
```

```
height:300px;
background:#990000;
float:right;
clear:left;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000;
}
```

秘笈心法

心法领悟 198：使数据居中显示的方法。

CSS 样式中有一个 text-align 属性，相当于使用<table>标签中的 center 属性，可以使页面数据居中显示。

实例 199	页面布局（2 列右侧固定左侧自适应宽度） 光盘位置：光盘\MR\03\199	高级 趣味指数：★★★★

实例说明

灵活地控制页面中<div>标签的宽度和高度是使用<div>标签的核心。本实例实现身体部分 2 列右侧固定左侧自适应宽度，未知高度+头部+导航+尾部的页面布局，如图 3.55 所示。

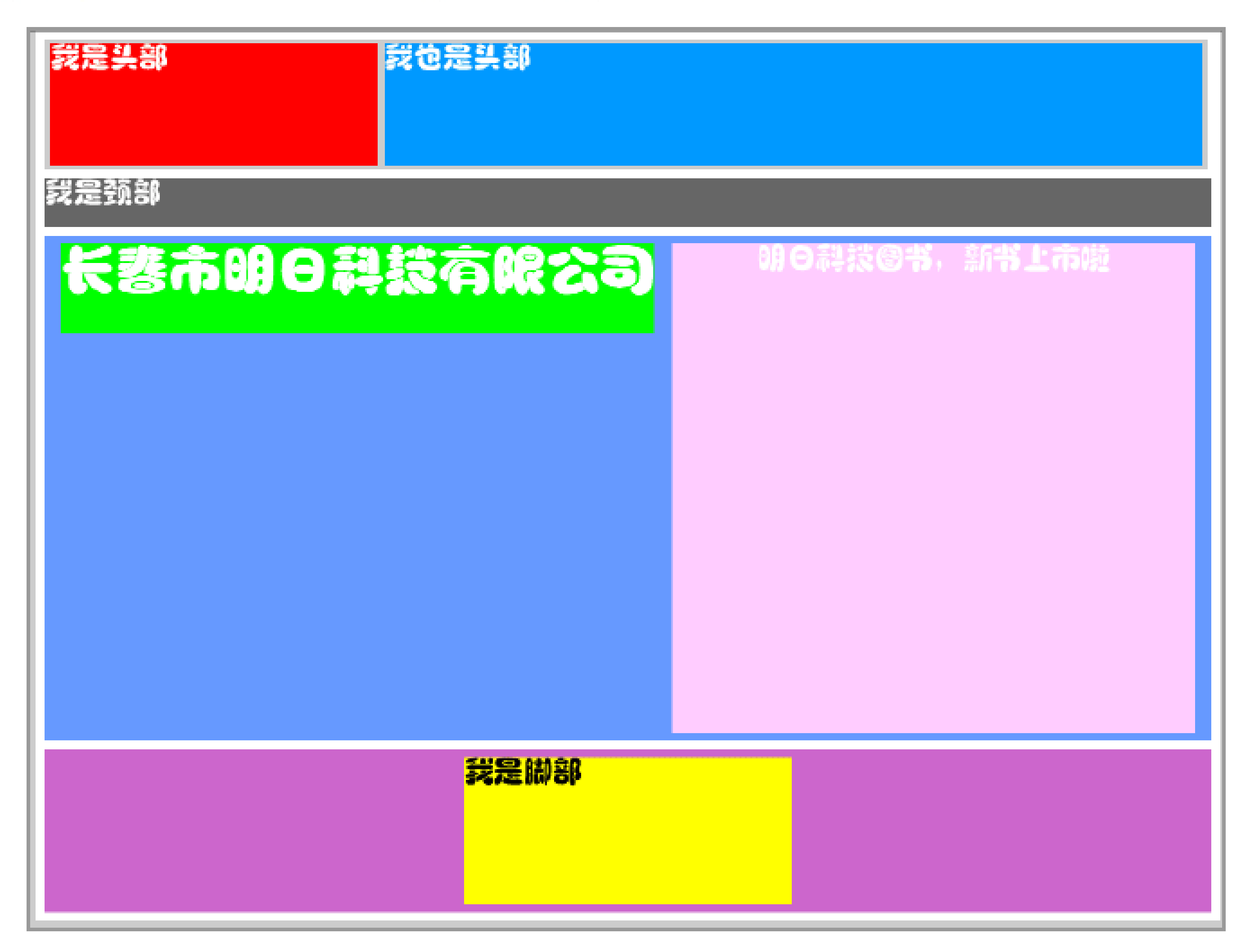


图 3.55　页面布局之 2 列右侧固定左侧自适应宽度

关键技术

本实例的关键点是实现身体部分左侧自适应宽度且未知高度。实现思想是设置 width、height 属性，代码

如下：

```
.g{
margin:5px;
width:auto;
height:auto;
background:#00FF00;
float:left;
}
```

设计过程

（1）创建脚本文件，利用<div>标签建立网页的主体架构，输出文本信息，代码如下：

```
<div class="a">
<div class="b">
	<div class="f">我是头部</div>
	<div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
	<div class="g">
		<h1>长春市明日科技有限公司</h1>
	</div>
	<div class="h">
		明日科技图书，新书上市啦
	</div>
</div>
<div class="e">
	<div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的具体属性，并更改输出文本的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
```

```
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:auto;
height:auto;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:320px;
height:300px;
background:#FFCCFF;
float:right;
text-align:center;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000;
}
```

秘笈心法

心法领悟 199：关键字 auto 的使用。

在<div>标签中，如果不知道该标签的具体宽度和长度或者与四边的距离，可以将这些属性设置为 auto。这样可以让<div>标签的长度和宽度自动调节。

实例 200	页面布局（2 列左侧固定右侧自适应宽度+底平齐） 光盘位置：光盘\MR\03\200	高级 趣味指数：★★★★

实例说明

利用<div>设计页面时，有很多很小的技巧，灵活地掌握这些技巧可以使用户对页面设计豁然开朗。本实例主要实现身体部分的 2 列左侧固定右侧自适应宽度，未知高度且底部平齐+头部+导航+尾部的页面布局，运行效果如图 3.56 所示。

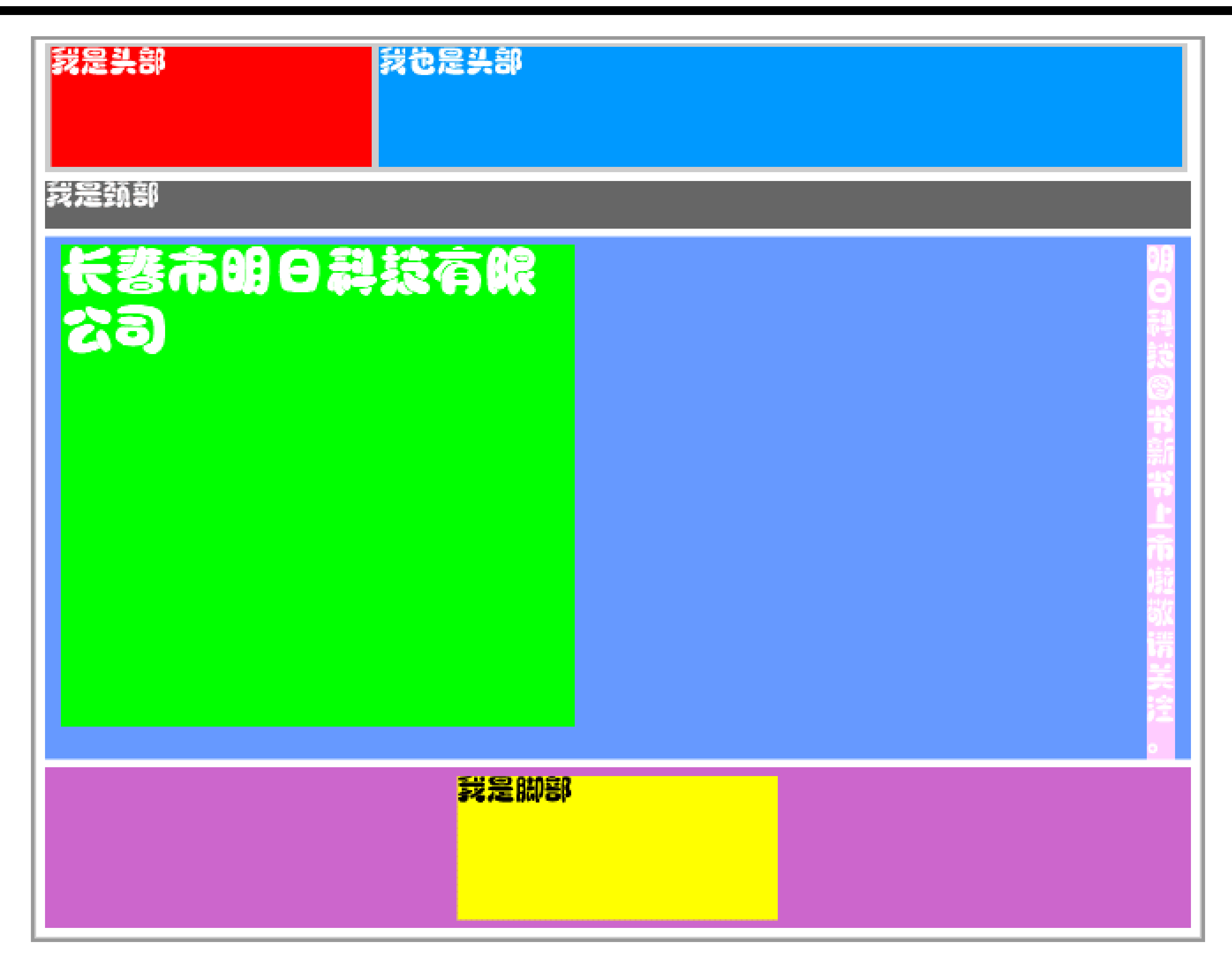


图 3.56　页面布局之 2 列左固定右自适宽度+底平齐

关键技术

本实例的关键点是实现身体部分右侧自适应宽度且未知高度，底部平齐。实现思想是设置 width、height、margin 属性，代码如下：

```
.h{
margin:5px 5px 0px 5px;
width:auto;
height:auto;
background:#FFCCFF;
float:right;
text-align:center;
}
```

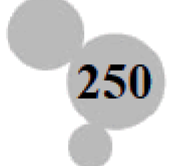
设计过程

（1）创建脚本文件，利用<div>标签建立网页的主体架构，输出文本信息，代码如下：

```
<div class="a">
<div class="b">
        <div class="f">我是头部</div>
        <div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
        <div class="g">
                <h1>长春市明日科技有限公司</h1>
        </div>
        <div class="h">
                明<br>日<br>科<br>技<br>图<br>书<br>新<br>书<br>上<br>市<br>啦<br>敬<br>请<br>关<br>注<br>
        </div>
</div>
<div class="e">
        <div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的具体属性，并更改输出文本的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
```



```
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:320px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px 5px 0px 5px;
width:auto;
height:auto;
background:#FFCCFF;
float:right;
text-align:center;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000;
}
```

秘笈心法

心法领悟 200：属性 margin 与底部平齐。

margin 属性在实例 185 中已经讲解过，所谓的底部平齐就是指下边距为 0 像素，所以只要设置当前模块的下边距属性值为 0 像素即可。

实例 201	页面布局（2 列右侧固定左侧自适应宽度+底平齐） 光盘位置：光盘\MR\03\201	高级 趣味指数：★★★★

实例说明

本实例是实例 200 的对照，意在使读者对<div>定位技术有深入的了解，运行结果如图 3.57 所示。

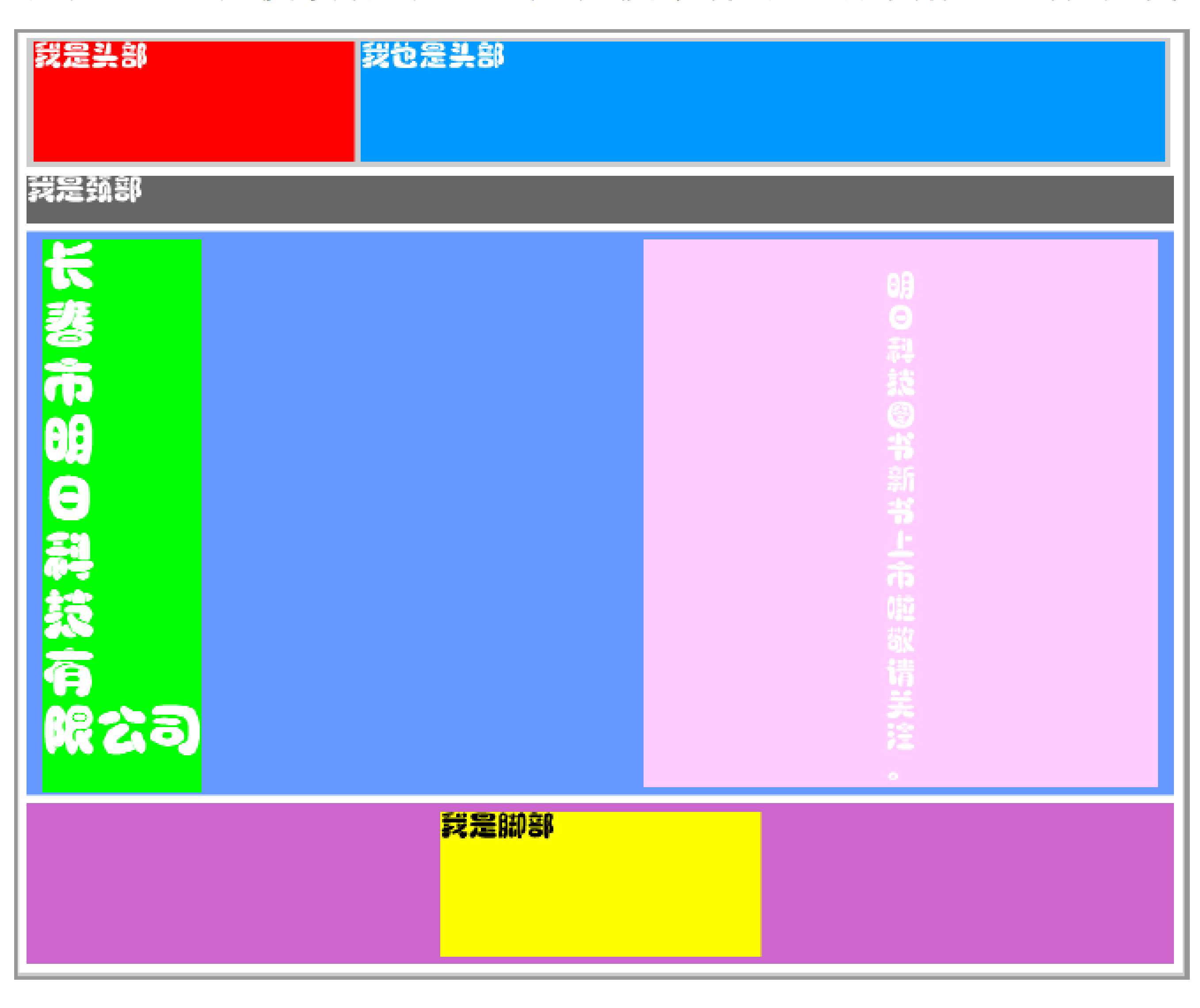


图 3.57　页面布局之 2 列右固定左自适宽度+底平齐

关键技术

本实例的方法与实例 200 相同，这里不再赘述。

设计过程

（1）创建脚本文件，利用<div>标签建立网页的主体架构，输出文本信息，代码如下：

```
<div class="a">
<div class="b">
        <div class="f">我是头部</div>
        <div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
        <div class="g">
                <h1>长春市明日科技有限公司</h1>
        </div>
        <div class="h">
```



```
            明<br>日<br>科<br>技<br>图<br>书<br>新<br>书<br>上<br>市<br>啦<br>敬<br>请<br>关<br>注<br>
        </div>
</div>
<div class="e">
        <div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的具体属性，并更改输出文本的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px 5px 0px 5px;
width:auto;
height:auto;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:320px;
height:300px;
background:#FFCCFF;
```

```
float:right;
text-align:center;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000;
}
```

秘笈心法

心法领悟 201：padding 补丁边距。

本实例中，因为设置底部平齐，所以右侧固定宽度的模块就可能与底边距不是 5 个像素的距离，为了整体的美感效果，可以为右侧模块设置 padding 补丁边距，设置好合适距离后，显示效果如图 3.57 所示。

3.6 弹 出 窗 口

弹出窗口经常在网站中使用，如弹出式广告或网站公告及打开居中显示详细信息的新窗口等。所以掌握设计弹出窗口的技术，也许会给设计的网站带来额外的效益。

实例 202	弹出指定尺寸的无边框窗口 光盘位置：光盘\MR\03\202	高级 趣味指数：★★★★

实例说明

JavaScript 这门脚本语言的功能很强大，尤其是在网页特效的处理上更是有其独到之处。本实例将讲述的是通过 JavaScript 代码控制，弹出指定尺寸的无边框窗口，如图 3.58 所示。



图 3.58 单击跳转图



关键技术

本实例主要是利用 JavaScript 代码，控制弹出窗口的尺寸，核心代码如下：

```
self.resizeTo(250,180);
width=screen.width;
height=screen.height;
self.moveTo((width-240)/2,(height-139)/2);
```

设计过程

（1）创建脚本文件，利用<div>标签定位技术建立网页架构，引入 CSS 文件，当单击图标按钮时发生跳转，代码如下：

```
<div class="a"><img src="pic/bg.jpg" border="0" usemap="#Map" />
            <map name="Map" id="Map"><area shape="rect" coords="469,298,541,382" href="in.php" /></map>
</div>
```

（2）创建脚本文件，命名为 in.php，此文件执行跳转后的代码。另外，在此文件中建立 form 表单，代码如下：

```
<div class="c">
      用户名：<input type="text" name="text"><br>
      密  码：<input type="password"><br>
      <input type="button" class="one" value="">  <input type="button" class="two"value="">
</div>
```

（3）编写 JavaScript 脚本，定义网页窗口的尺寸并定义当鼠标覆盖和移开文本框时产生的变色效果，代码如下：

```
$(document).ready(function(){
self.resizeTo(250,180);                                    //定义窗口的宽和高
width=screen.width;
height=screen.height;
self.moveTo((width-240)/2,(height-139)/2);
$("input:eq(0)").mouseover(function(){                     //当鼠标覆盖时
      $("input:eq(0)").css("background","#FFCCFF");
      $("input:eq(0)").mouseout(function(){                //当鼠标移开时
            $("input:eq(0)").css("background","#FFFFFF");
      });
});
$("input:eq(1)").mouseover(function(){
      $("input:eq(1)").css("background","#FFCCFF");
      $("input:eq(1)").mouseout(function(){
            $("input:eq(1)").css("background","#FFFFFF");
      });
});
});
```

秘笈心法

心法领悟 202：HTML 的<map>标签。

如果用户使用的编程工具是 Dreamweaver，那么对于<map>标签应该并不陌生，此标签的主要作用是当鼠标单击指定区域时，网页发生跳转。

实例 203　全屏显示无边框有滚动条窗口

光盘位置：光盘\MR\03\203

高级

趣味指数：★★★★

实例说明

通常用户在使用 IE 浏览器浏览网页时，浏览器窗口包括标题栏、菜单栏和状态栏等固定内容。虽然菜单栏和状态栏可以通过设置而隐藏，但是标题栏却不能隐藏，即使将网页全屏显示也不能将其去除。有时为了网站

的整体效果考虑，需要将网页设置为全屏显示。例如，在明日网络宽带影院网站中，为了给用户以更真实的感觉，就可以将整个网站的主页面设置为全屏显示模式，如图 3.59 所示，从图中可以看出该页面中还设计了“退出”超链接，可以方便浏览者随时离开影院，这也是设计全屏显示时需要注意的问题，否则浏览者会因找不到出口而烦恼。



图 3.59　全屏显示

关键技术

本实例主要通过 window 对象的 open()方法，并借助一个中转页实现，当中转页运行时调用 open()方法打开一个全屏显示的窗口（使用 window 对象的 open()方法中的参数 fullscreen=1 实现），并关闭打开中转页的窗口。

设计过程

新建一个空的 PHP 页面（页面中可以没有任何页面布局元素或内容），命名为 index.php。该页面只是起到一个页面跳转的作用，主要应用 JavaScript 脚本自定义一个 MM_c()函数，并应用 window 对象的 open()方法打开一个新的 formal.htm 页，设置参数 fullscreen=1，从而实现打开窗口的全屏显示，代码如下：

```
<object type="application/x-oleobject" id=closes classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11">
<param name="command" value="close">
</object>
<script language="javascript">
MM_c();
function MM_c(){
            closes.Click();
            window.open("formal.htm","","fullscreen=1");
}
</script>
```

秘笈心法

心法领悟 203：全屏显示的主要应用。

用户在浏览类似于土豆、优酷等网站的视频时，经常会看到播放器的下方存在一个全屏显示的按钮，其实现方法与本实例大同小异，只是使用的控制方法不同而已。

实例 204	网页拾色器 光盘位置：光盘\MR\03\204	高级 趣味指数：★★★★

实例说明

在网站开发过程中，用户经常要求在客户端应用自定义颜色来改变某些信息或控件等的颜色值，如留言本中的文字颜色、控件背景色等，通过在网页中加入一个网页拾色器即可实现。运行本实例，将在页面中显示如图 3.60 所示的网页拾色器，当用户单击颜色块时，就会弹出一个提示框，提示其所选的颜色值是多少，通过选择的颜色值来改变网页的背景色。



图 3.60　网页拾色器

关键技术

本实例的关键点是在实现网页拾色器时，应用 JavaScript 的数组。创建数组有以下 3 种方法：

（1）无参数调用，语法格式如下：

```
var h = new Array();
```

（2）指定数组前 n 个元素的值，语法格式如下：

```
var h = new Array(arglist);
```

其中参数 arglist 是一个用逗号隔开的值表，这些值用于给 variant 所包含的数组的各元素赋值。如果不提供参数，则创建一个长度为 0 的数组。

（3）指定数组的元素个数，语法格式如下：

```
var h = new Array(n);
```

其中参数 n 是指定数组的长度。由于在 JavaScript 中，数组的第 1 个元素的下标值为 0，所以 n 的值为数组的最大下标值加 1。

设计过程

（1）创建需要调用网页拾色器的页面，在该页面中添加一个文本框，将文本框设置为只读，同时将文本框的背景色设置为黑色，并在该文本框的 onClick 事件中调用自定义的 JavaScript 函数 colorpick()，代码如下：

```
<input name="color" type="text" id="color" size="3" readonly="yes" style="background-color:#CCFF00" onClick="colorpick(this);">
```

（2）编写自定义的 JavaScript 函数 colorpick()，用于打开网页对话框调用网页拾色器，并用网页拾色器的

返回值设置文本框的背景色和网页的背景色，代码如下：

```
<script language="javascript">
function colorpick(field){
    var rtn = window.showModalDialog("color.php","","dialogWidth:225px;dialogHeight:170px;status:no;help:no;scrolling=no;scrollbars=no");
    if(rtn!=null)
 document.body.style.background=rtn;
 field.style.background=rtn;
        return;
    }
</script>
```

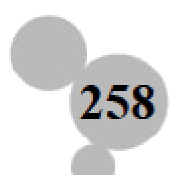
（3）制作网页拾色器页面 color.php，关键代码如下：

```
<script language="JavaScript">
var h = new Array(6)
h[0] = "FF";
h[1] = "CC";
h[2] = "99";
h[3] = "66";
h[4] = "33";
h[5] = "00";
function action(RGB) {
  parent.window.returnValue="#"+RGB;
    window.close();
}
function Mcell(R, G, B) {
  document.write('<td bgcolor="#' + R + G + B + '">');
  document.write('<a href="#" onClick="action(\'' + (R + G + B) + '\')">');
  document.write('<img border=0 height=12 width=12 \')" alt=\'#'+R+G+B+'\'>');
  document.write('</a>');
  document.write('</td>');
}
function Mtr(R, B) {
  document.write('<tr>');
  for (var i = 0; i < 6; ++i) {
   Mcell(R, h[i], B);
  }
  document.write('</tr>')
}
function Mtable(B) {
  document.write('<table cellpadding=0 cellspacing=0 border=0>');
  for (var i = 0; i < 6; ++i) {
   Mtr(h[i], B);
  }
document.write('</table>');
}
function Mcube() {
  document.write('<table cellpadding=0 cellspacing=0 border=0><tr>');    //
  for (var i = 0; i < 6; ++i) {
    if(i%3==0){
    document.write('<tr>');
    }
   document.write('<td bgcolor="#FFFFFF">');
   Mtable(h[i])
   document.write('</td>');
  }
    if(i%3==0){
    document.write('</tr>');
    }
  document.write('</tr></table>');
}
Mcube()
</script>
```

秘笈心法

心法领悟 204：浅谈 variant 变量。

本实例中没有作为数组声明的 variant 也可以表示数组。除了长度固定的字符串以及用户定义类型之外，



variant 变量可以表示任何类型的数组。尽管一个包含数组的 variant 和一个元素为 variant 类型的数组在概念上有所不同，但对数组元素的访问方式是相同的。

实例 205　日期选择器

光盘位置：光盘\MR\03\205

高级

趣味指数：★★★★

实例说明

目前，日期型数据格式有多种，采用录入方式相对来说比较繁琐，而且采用这种方式也不利于日期格式的统一，所以可以在信息录入页面中加入一个简单的日期选择器来解决上述问题。运行本实例，在新奥家电连锁后台管理系统的销售查询页面中，单击“售货日期”文本框后的日期选择按钮，会弹出“日期选择器”对话框，选择售货日期的起始日期后，单击“确定”按钮，即可成功地将选择的日期添加到对应的日期文本框中，运行结果如图 3.61 所示。



图 3.61　日期选择器

关键技术

由于日期为连续的数字，开发网页时将这些信息逐个写在<option>与</option>标记之间固然可行，但这样会增加程序开发时间，在开发本实例时，笔者将通过 PHP 的循环结构显示出下拉列表框中的连续数据，这样更灵活方便，实现该过程的代码如下：

```
<form name="form2">
  <tr bgcolor="#826650">
    <td width="13"><div align="center"><a href="#" onClick="addday()"></a></div></td>
    <td width="80">
<select name="n1">
<?php
 for($i=2005;$i<=2050;$i++){
?>
      <option value=<?php echo $i;?>><?php echo $i;?></option>
<?php
  }
?>
    </select>
    年</td>
    <td width="64"><select name="y1">
<?php
 for($i=1;$i<=12;$i++){
?>
      <option value=<?php echo $i;?>><?php echo $i;?></option>
  <?php
  }
```

```
?>
      </select>
      月</td>
      <td width="77"><select name="r1">
<?php
 for($i=1;$i<=31;$i++){
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
 <?php
   }
?>
      </select>
      日</td>
      <td width="16"><div align="center"><a href="#" onClick="subday()"></a></div></td>
   </tr>
   </form>
```

设计过程

（1）在父窗口中创建函数 open_day_from()和 open_day_to()，实现子窗口的弹出和父窗口与子窗口之间的信息传递，代码如下：

```
<table width="250" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
 <form name="form2">
   <tr bgcolor="#826650">
      <td width="13"><div align="center"><a href="#" onClick="addday()"></a></div></td>
      <td width="80">
<select name="n1">
<?php
 for($i=2005;$i<=2050;$i++) {
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
<?php
   }
?>
      </select>
      年</td>
      <td width="64"><select name="y1">
<?php
 for($i=1;$i<=12;$i++) {
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
   <?php
   }
?>
      </select>
      月</td>
      <td width="77"><select name="r1">
<?php
 for($i=1;$i<=31;$i++) {
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
 <?php
   }
?>
      </select>
      日</td>
      <td width="16"><div align="center"><a href="#" onClick="subday()"></a></div></td>
   </tr>
   </form>
</table>
<table width="250" height="20" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
      <td bgcolor="#826650"><div align="center"><input type="button" onClick="close_day()" class="buttoncss" value="确定"></div></td>
   </tr>
</table>
```



（2）单击“确定”按钮，调用 close_day()函数实现日期的传值，代码如下：

```
<script language="javascript">
function close_day(){
window.returnValue=document.form2.n1.value+"-"+document.form2.y1.value+"-"+document.form2.r1.value;
   window.close();
}
</script>
```

秘笈心法

心法领悟 205：JavaScript 弹出窗口。

实现窗口之间的信息传递主要通过 window 的 showModalDialog()方法实现，其语法格式如下：

```
window.showModalDialog('弹出窗口地址', '弹出窗口名称', '弹出窗口所具有的属性')
```

窗口所具有的属性包括窗口的高度、宽度，以及窗口是否有地址栏、状态栏、滚动条等一些特性。

实例 206　弹出全屏显示的网页模式对话框

光盘位置：光盘\MR\03\206

高级

趣味指数：★★★★

实例说明

为了方便用户浏览网页信息，可以将网页信息以全屏显示的网页模式展示给用户，这样一来可以方便用户浏览到网页的全部信息；二来可以保证用户查看后能及时将其关闭。运行本实例，在明日科技编程词典网站中单击“购买须知”超链接后，将弹出如图 3.62 所示的购买须知页面，该页面就是通过全屏显示的网页模式对话框，用户只有关闭该页面后，才可以查看网站的其他内容。

★ 编程词典系列版本及价格 ★

● Visual Basic编程词典

软件版本	功能	享受服务	价格（元）
标准版	详细功能	一年两次升级	128
开发版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供30个项目源程序的有限服务支持。	568
专业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供60个项目源程序的有限服务支持。	1998
企业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。提供一年全程技术支持服务。	6998

● Delphi编程词典

软件版本	功能	享受服务	价格（元）
标准版	详细功能	一年两次升级	128
开发版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供30个项目源程序的有限服务支持。	568
专业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供60个项目源程序的有限服务支持。	1998
企业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。提供一年全程技术支持服务。	6998

图 3.62　弹出全屏显示的网页模式对话框

关键技术

本实例主要应用 screen 对象的 width、height 属性和 window 对象的 showModalDialog()方法实现，showModalDialog()方法用于弹出全屏显示的网页模式对话框，其语法格式如下：

```
variant = object.showModalDialog(sURL [, vArguments [, sFeatures]])
```

参数说明：

sURL：指定 URL 文件地址。

vArguments：用于向网页对话框传递参数。传递参数的类型不受限制，对于字符串类型，最大为 4096 个字

符，但也可以传递对象，如 index.html。

设计过程

（1）创建提供“购买须知”超链接的页面，在该页面中添加空的超链接，在该超链接的 onClick 事件中调用自定义 JavaScript 函数 opendialog()，代码如下：

```
<a href="#" onClick="opendialog()">购买须知</a>
```

（2）编写自定义 JavaScript 函数 opendialog()，用于打开全屏显示的网页对话框，代码如下：

```
<script language="javascript">
function opendialog(){
var width=screen.width;
var height=screen.height;
window.showModalDialog("notice.php","","dialogWidth="+width+"px;dialogHeight="+height+"px;status=no;help=no;scrollbars=no")
}
</script>
```

秘笈心法

心法领悟 206：showModalDialog()与 showModelessDialog()的区别是什么。

二者的区别在于 showModalDialog()打开的网页对话框为模式窗口，置于父窗口上，必须关闭才能访问父窗口；而 showModelessDialog()打开的对话框是无模式窗口，打开后不必关闭也可以访问父窗口或其他窗口。

实例 207　关闭 IE 主窗口时不弹出询问对话框

光盘位置：光盘\MR\03\207

高级

趣味指数：★★★★

实例说明

通常情况下，当使用 JavaScript 的 window.close()语句关闭 IE 主窗口时，会弹出一个“您查看的网页正在试图关闭窗口。是否关闭此窗口？”的询问对话框。在制作网络应用程序时，这种情况是用户不想看到的，那么如何才能屏蔽该对话框呢？本实例将介绍解决该问题的方法。运行本实例，效果如图 3.63 所示，单击“关闭窗口”超链接后，该 IE 窗口将被直接关闭。



图 3.63　关闭窗口不弹出对话框

关键技术

本实例主要应用 window 对象的 opener 属性将要关闭的 IE 窗口的打开窗口设置为 null，然后再通过 window 对象的 close()方法将该窗口关闭，这样就不会弹出询问对话框。

设计过程

创建 index.php 文件，添加一个用于关闭窗口的超链接，在该超链接的 onClick 事件中添加控制窗口关闭的



代码如下：

```
<a href="#" onClick="window.opener=null;window.close();">• 关闭窗口</a>
```

秘笈心法

心法领悟 207：详解 opener。

opener 是一个可读可写的属性，对一个 window 对象返回一个引用，用于指定打开本窗口的窗口对象，语法格式如下：

```
window.opener
window.opener.方法
window.opener.属性
```

opener 属性返回的是一个窗口对象，其与打开该窗口的父窗口相联系，当访问子窗口的 opener 属性时，返回的是父窗口。通过该属性，可以使用父窗口对象中的方法和属性。

实例 208　弹出提示对话框并重定向网页

光盘位置：光盘\MR\03\208

高级

趣味指数：★★★★

实例说明

在网站开发过程中，需要考虑网站的安全性，例如，设置允许哪些用户访问是一项非常重要的技术。本实例在页面中设置一个访问权限，判断用户是否可以访问该网页。如果用户名和密码不正确就不可以访问，并弹出提示对话框重定向到该网页，运行结果如图 3.64 所示。

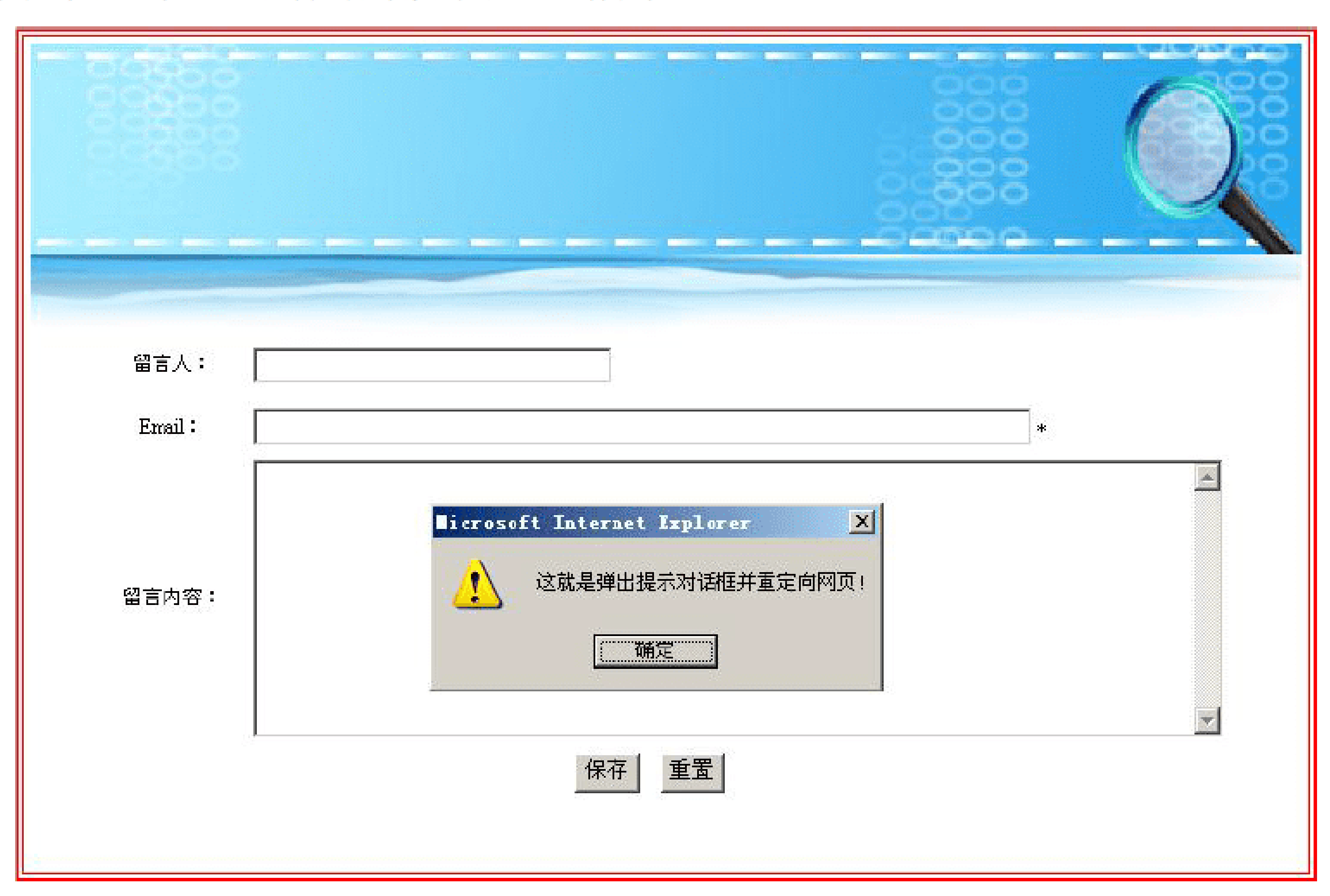


图 3.64　弹出提示对话框并重定向网页

关键技术

本实例主要将 PHP 与 JavaScript 相结合，通过 JavaScript 脚本来弹出对话框信息，应用 window 对象中的 alert() 方法和窗口属性 location 实现此功能，语法格式如下：

```
alert ('弹出对话框的内容');
window.location.href='重新定向的页面';
```

参数说明：

alert()：弹出一个警告对话框。

location：定位区，也叫地址栏，是可以输入 URL 的浏览器文本区。

设计过程

弹出提示对话框并重定向网页文件的关键代码如下：

```
<?php session_start();?>
<title>弹出提示对话框并重定向网页</title>
<?php
if($gly==true && $mima==true){
if($ljid==true){
$ljid=$_GET['ljid'];
echo "这就是弹出提示对话框并重定向网页!";}
?>
<?php }else{
echo "<script>alert ('这就是弹出提示对话框并重定向网页!'); window.location.href='admin.php';</script>";
}
?>
```

秘笈心法

心法领悟 208：浅谈网页重定向。

这里的网页重定向与 PHP 判断用户登录权限页相似，都是网页的一种跳转方法。

实例 209	自动关闭的广告窗口 光盘位置：光盘\MR\03\209	高级 趣味指数：★★★★

实例说明

进入网站后，很多网站都会弹出一个新窗口（如广告等），多数窗口需要浏览者自行关闭。为了方便浏览者对页面中信息的浏览，本实例运用一种新的方法来解决弹出窗口关闭这一问题。当浏览者进入某网站进行浏览时，无须关闭弹出的新窗口，在页面运行超过一定的时间之后，该窗口将自动关闭，这样将大大方便浏览者的浏览。运行本实例，将弹出一个新窗口显示广告信息，如图 3.65 所示，该广告窗口在页面运行 5 秒钟后自动关闭。



图 3.65　自动关闭的广告窗口

关键技术

本实例主要应用 window 对象的 setTimeout()方法，实现延迟执行某一操作。window 对象的 setTimeout()方法的语法格式如下：

```
setTimeout(expression,secdelay[,language])
```



参数说明：

expression：是一个字符串，可包含任何对函数、方法或者单个 JavaScript 语句的调用。

secdelay：指定运行间隔的时间，以毫秒为单位。

language：指定语句或参数 expression 调用的函数所使用的脚本语言。如果完全使用 JavaScript，则不需设置该参数。

设计过程

（1）通过 JavaScript 脚本实现打开新窗口功能，代码如下：

```
<script language="javascript">
window.open("new.htm","new","height=135,width=180,top=10,left=20");
</script>
```

（2）通过设置 window 对象的 setTimeout()方法，实现弹出新窗口的自动关闭，代码如下：

```
<body onLoad="window.setTimeout('window.close()',5000)">
```

秘笈心法

心法领悟 209：秒与毫秒。

如果用户不是运动员，很少能接触到毫秒这个单位，但是在计算机领域中，毫秒是经常被用到的。秒与毫秒的换算比例是 1:1000，也就是说 1 秒等于 1000 毫秒。

实例 210	在弹出的对话框中选择个性头像 光盘位置：光盘\MR\03\210	高级 趣味指数：★★★★

实例说明

在论坛、博客等注册页面添加头像选择功能，可以增加网页活力。运行本实例，首先单击图 3.66 中的“头像选择”超链接，在弹出的窗口中选择自己喜爱的头像后，弹出窗口将自动关闭，注册页面中的头像变成用户所选择的头像。



图 3.66　选择头像

关键技术

本实例通过调用 window 对象的 showModalDialog()方法弹出一个窗口，当用户单击某个头像后，将该头像对应的数值传递给父窗口，父窗口通过传递过来的数值指定用户所选择的头像。



设计过程

（1）首先在窗口中添加“头像选择”超链接，以便调用 select_face()函数实现窗口的弹出功能，代码如下：

```
[<a href="#" onclick="select_face()" class="a1">头像选择</a>]
```

（2）利用 JavaScript 编写 select_face()函数，实现头像选择窗口的弹出以及返回用户所选择头像对应的数值，代码如下：

```
<script language="javascript">
function select_face(){
var somevalue;
somevalue=window.showModalDialog("browse.php","请选择您喜欢的人物头像", "dialogWidth='310px';dialogHeight='240px';status='no';help='no';
scrollbars='no'");
document.form regist.user src.src="images/head/"+somevalue+".gif";
 }
</script>
```

（3）在弹出的窗口中通过 selectface()函数返回用户所选择头像对应的数值，代码如下：

```
<script language="javascript">
function selectface(ICQ){
window.returnValue=ICQ;
window.close();
}
</script>
```

秘笈心法

心法领悟 210：showModalDialog()方法详解。

```
window.showModalDialog('弹出窗口地址', '弹出窗口名称', '弹出窗口所具有的属性')
```

窗口所具有的属性包括窗口的高度、宽度，以及窗口是否有地址栏、状态栏、滚动条等一些特性。

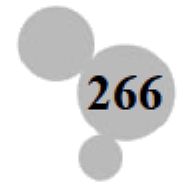
实例 211	浮动的广告窗口 光盘位置：光盘\MR\03\211	高级 趣味指数：★★★★

实例说明

网络不仅是信息的海洋，也是广告的海洋。除了普通的 gif 和 flash 之外，浮动广告也是时下网上较为流行的广告形式之一，当拖曳浏览器的滚动条时，这种页面上浮动的广告，可以跟随屏幕一起移动，这种效果对于广告展示有相当大的实用价值。本实例将制作一个浮动的广告窗口，运行结果如图 3.67 所示。



图 3.67　浮动的广告窗口



关键技术

本实例主要应用 DIV 层制作浮动广告，然后应用 document.write()方法输出 DIV 层。document.write()方法的语法格式如下：

```
document.write(value,…)
```

参数说明：

value：要添加到 document 的任意一个 JavaScript 值。如果该值不是字符串，那么在被添加至文档之前，会被转化成一个字符串。

…：任意多个（0 个或多个）要（依次）写入文档的值。

设计过程

（1）在网站首页引入 JavaScript 脚本文件，代码如下：

```
<SCRIPT language=JavaScript src="js/left.js"></SCRIPT>
```

（2）运用 JavaScript 脚本自定义函数，控制浮动广告的位置及其相关属性，代码如下：

```
var adve=0.15
var collection;
function motion() {
this.items= [];
this.addItem= function(id,x,y,content){
document.write('<DIV id='+id+' style="Z-INDEX: 10; POSITION: absolute;    width:80px;
height:60px;left:'+(typeof(x)=='string'?eval(x):x)+';top:'+(typeof(y)=='string'?eval(y):y)+'">'+content+'</DIV>');
var newItem= {};
newItem.object= document.getElementById(id);
newItem.x= x;
newItem.y= y;
this.items[this.items.length]= newItem;
  }
this.play=function(){
collection=this.items
setInterval('play()',10);
  }
}
function play()        {
for(var i=0;i<collection.length;i++)        {
var followOb= collection[i].object;
var followObj_x     = (typeof(collection[i].x)=='string'?eval(collection[i].x):collection[i].x);
var followObj_y= (typeof(collection[i].y)=='string'?eval(collection[i].y):collection[i].y);
if(followObj.offsetLeft!=(document.body.scrollLeft+followObj_x)) {
var dx=(document.body.scrollLeft+followObj_x-followObj.offsetLeft)*adve;
dx=(dx>0?1:-1)*Math.ceil(Math.abs(dx));
followObj.style.left=followObj.offsetLeft+dx;
}
if(followObj.offsetTop!=(document.body.scrollTop+followObj_y)) {
var dy=(document.body.scrollTop+followObj_y-followObj.offsetTop)*adve;
dy=(dy>0?1:-1)*Math.ceil(Math.abs(dy));
followObj.style.top=followObj.offsetTop+dy;
}
followObj.style.display     = '';
}
}
var themotion= new motion();
themotion.addItem('followDiv1',200,200,'<a href="http://wpa.qq.com/msgrd?V=1&Uin=228691050&Site=在线咨询&Menu=no" title="在线即时交谈
"><img src=./images/qqhelp.gif    height=70 width="80" border=0></a>');
themotion.play();
```

秘笈心法

心法领悟 211：使用 document.write()方法的注意事项。

调用 document.write()方法的结果也许不能立刻在目标浏览器窗口中显示出来，这是因为浏览器可能会将数据缓存起来，以便形成大的数据块输出。

实例 212 在页面右下角弹出渐显的广告窗口

光盘位置：光盘\MR\03\212

高级

趣味指数：★★★★

实例说明

广告作为网站最大的盈利手段，任何网站都不可能将其省略，那么，如何才能实现网站既可以从广告中获利，又能使用户流畅地阅读网站信息呢？本实例将介绍如何制作一个从首页右下角弹出的渐显广告窗体，该窗体在用户登录网站时自动从右下角渐渐弹出，然后由用户手动关闭。本实例的运行结果如图 3.68 所示。



图 3.68　右下角渐显广告

关键技术

本实例主要应用 JavaScript 脚本的 window 对象，window 对象主要用于在 HTML 中打开窗口，应用极为普遍，但也有一些缺陷。用户浏览器决定窗口的外观，设计者左右不了其窗口的大小及样式，但 JavaScript 给了程序这种控制权。在 JavaScript 中，可以使用 window 对象来实现对窗口的控制。

设计过程

本实例主要应用 window 对象的 open()方法打开一个在首页右下角弹出渐显的广告窗口，在网站首页中添加如下代码：

```
<script language="javascript">
var newformW=300;
var newformH=180;
function pp(){
 var T=window.screen.width-newformW;
 var L=window.screen.height+newformH;
 var name=window.open("advertise.htm","","width="+newformW+",height="+newformH+",top="+T+",left="+L);
}
pp();
</script>
```

秘笈心法

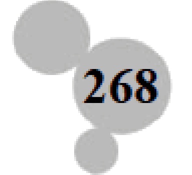
心法领悟 212：window 对象打开窗口详解。

使用 window 对象打开窗口的语法格式如下：

```
windowVar=window.open(url,windowname[,location]);
```

参数说明：

windowVar：当前打开窗口的句柄。如果 open()方法成功，则 windowVar 的值为一个 window 对象的句柄，



否则 windowVar 的值是一个空值。

url：目标窗口的 URL。如果 URL 是一个空字符串，则浏览器将打开一个空白窗口，允许用 write()方法创建动态 HTML。

windowname：window 对象的名称。

location：对窗口属性进行设置。

3.7　状态栏设计

不论是在应用程序还是在网页中都会应用到状态栏，状态栏可为用户提供一些提示信息，如当前系统的日期和时间、软件版本、链接地址、系统信息以及一些自定义信息等。

实例 213	跑马灯文字 光盘位置：光盘\MR\03\213	高级 趣味指数：★★★★

实例说明

浏览网页时，用户往往更加注意网站页面的内容，而不会过多注意状态栏，如果为页面的状态栏添加一些特效，一定会给网站增添一道亮丽的风景。本实例实现在状态栏中显示跑马灯特效文字，运行结果如图 3.69 所示。



图 3.69　跑马灯文字

关键技术

本实例主要应用 JavaScript 脚本的 window 对象的 status 属性，并通过编写自定义函数设置状态栏中的文字及文字属性，通过 if...else...条件语句控制文字的循环显示效果。

最后，在<body>标记中调用该函数，从而实现状态栏中的跑马灯文字特效。

设计过程

（1）利用 JavaScript 脚本编写自定义函数，生成状态栏中的文字及文字属性，代码如下：

```
<SCRI PT language="JavaScript">
function scrollit(s){
var text="      您好！欢迎访问明日科技图书网站。客服热线：0431-84978981"
var printtext = " ";
var i = 1;
if (s > 210){
```

```
s--;
var out="scrollit("+s+")";
var timerTwo=window.setTimeout(out,200);
}
else if (s <= 210 && s > 0){
for (i=0;i<s;i++){
printtext+=" ";
}
printtext+=text;
s--;
window.status=printtext;
out="scrollit("+s+")";
timerTwo=window.setTimeout(out,200);
}
else if (s <= 0){
if (-s < text.length){
printtext+=text.substring(-s,text.length);
s--;
window.status=printtext;
out="scrollit("+s+")";
timerTwo=window.setTimeout(out,200);
}
else{
window.status=" ";
timerTwo=window.setTimeout("scrollit(100)",200);
}
}
}
</SCRIPT>
```

（2）在<body>标记中调用自定义的函数。

```
<body onLoad="scrollit(100);">
```

秘笈心法

心法领悟 213：window 对象的 status 属性详解。

window 对象的 status 属性的语法格式如下：

```
window.status
```

status 属性是一个可读可写的字符串，声明浏览器状态栏的当前内容。

实例 214　状态栏中的收缩显示文字

光盘位置：光盘\MR\03\214

高级

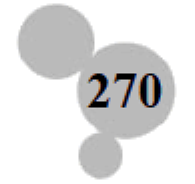
趣味指数：★★★★

实例说明

为了在节省网页空间的同时更好地服务于广大的用户，可以将网站要显示的简短的广告信息显示在状态栏中。运行本实例，在状态栏中收缩显示明日科技公司的网址和客服热线，运行结果如图 3.70 所示。



图 3.70　收缩显示文字



关键技术

本实例主要应用 window 对象的 status 属性。当显示的文字总长度等于该文本的长度时，再反方向滚动文字，从而实现在状态栏中收缩显示文字。

window 对象的 status 属性的语法格式如下：

```
window.status
```

属性 status 是一个可读可写的字符串，声明了浏览器状态栏的当前内容。

设计过程

本实例主要通过 JavaScript 脚本编写状态栏中收缩显示文字特效的自定义函数，完整代码如下：

```
<Script language=JavaScript>
var text1="欢迎访问明日科技主页：www.mingrisoft.com";
comeback=0;
cometo=0;
function dis(){
window.status=text1.substring(0,cometo);
if(comeback==1){
 cometo--;
 if(cometo==0){comeback=0}
}
else {cometo++;if(cometo==text1.length){comeback=1}}
window.status=text1.substring(0,cometo)+"(客服热线：0431-84978981)";
if(cometo==text1.length){window.setTimeout("dis()",1000)}
else {window.setTimeout("dis()",50)}
}
dis();
</Script>
```

秘笈心法

心法领悟 214：状态栏。

状态栏只是起到一个类似于广告的作用，但它比其他的弹出层广告逊色一些。

实例 215	状态栏中显示数字时钟 光盘位置：光盘\MR\03\215	高级 趣味指数：★★★★

实例说明

在状态栏中显示数字时钟是个性化网站所特有的功能，它不仅可以节约网页空间，而且可以方便用户时时查看。本实例实现在状态栏中显示一个 24 小时制的数字时钟，运行结果如图 3.71 所示。



图 3.71　状态栏显示时钟

关键技术

本实例主要应用 window 对象的 status 属性实现在状态栏中显示数字时钟，详细讲解请参考实例 214。

设计过程

（1）本实例主要通过 JavaScript 脚本开发状态栏中显示动态数字时钟的自定义函数，代码如下：

```
<script language="JavaScript">
var timerID = null;
var timerRunning = false;

function stopclock (){
      if(timerRunning)
            clearTimeout(timerID);
            timerRunning = false;
}

function showtime () {
      var now = new Date();
      var hours = now.getHours();
      var minutes = now.getMinutes();
      var seconds = now.getSeconds()
      var timeValue = "北京时间：" + ((hours >24) ? hours -24 :hours)
      timeValue += ((minutes < 10) ? ":0" : ":") + minutes
      timeValue += ((seconds < 10) ? ":0" : ":") + seconds
      window.status = timeValue;
      timerID = setTimeout("showtime()",1000);
      timerRunning = true;
}

function startclock () {
      stopclock();
      showtime();
}
</script>
```

（2）在<body>标记中调用自定义函数 startclock()，代码如下：

```
<body onLoad="startclock()">
```

秘笈心法

心法领悟 215：JavaScript 脚本中时间的设置。

在 JavaScript 脚本中设置日期和时间，首先要实例化 Date 类，然后调用其中的方法，获取对应的日期、时间值。

3.8 导航菜单设计

网站导航菜单对于每个网站都是必不可少的，它相当于城市中的路标，可以指引浏览者找到自己热衷的内容。导航菜单有很多种，如本节中介绍的带图标的文字导航条、Flash 导航条、按钮导航条、弹出式下拉菜单、二级导航菜单、树状菜单等。

实例 216	树状导航菜单 光盘位置：光盘\MR\03\216	高级 趣味指数：★★★★

实例说明

树形结构能够以层次形式展示信息，用它来描述具有上下级关系的内容再恰当不过。本实例利用菜单的树



形结构来描述企业的人事组织架构，效果如图 3.72 所示。



图 3.72　树形导航菜单

关键技术

本实例中实现菜单的树形显示是通过<table>标签结合 DIV 实现的。父节点利用表格来显示，子节点利用 DIV 嵌套表格来实现。这样做的目的是容易对子节点的显示和隐藏进行控制。例如，当用户单击父节点时，如果子节点没有展开，将 DIV 设置为可见就可以显示子节点；如果子节点可见，将 DIV 设置为不可见就可以隐藏子节点。

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 2 行 1 列的表格，在表格的第 2 行再添加一个 2 行 2 列的表格。表格的第 1 列用于显示节点前的图标，第 2 列描述节点文本，主要代码如下：

```
<form id="form1" name="form1" method="post" action="" >
  <table width="372"  border="1" align="center"  bgcolor="#E6F2F2" bordercolor="#478D8D"
  style="border-style:none" cellspacing="0" >
    <tr>
      <td   scope="col" align="center" style="border-bottom-style:none" >人事组织架构</td>
    </tr>
    <tr>
      <td >
           <table width="100%"  border="0" cellpadding="0" cellspacing="0" align="left">
        <?php                                                          //嵌入 PHP 脚本

          $xml_file = simplexml_load_file("org.xml");                  //从 XML 文件中加载节点
              foreach($xml_file->children() as $node){                  //遍历根节点
      ?>
          <tr>
           <td width="38%" align="right">                               //设置节点图标
                 <img src="images/rootnode.gif" width="15" height="15" border="0" >   </img>
           </td>
            <td   width ="62%"   left="0px" align="left" ><a href="#" onclick= "ShowNode(<?php echo $node ?>)">  
           <?php echo $node ?></a>                                      //设置节点文本
            </td>
          </tr>
          <tr>
          <td width ="38%" align="center" >
          </td>
             <td   width ="62%" >
                   <div id= <?php echo $node ?> style = "display:none" class="HideDiv" >   //定义 DIV
                     <table width="100%" border="0" cellpadding="0" cellspacing="0" align="left" >
                         <?php
                           foreach($node->children() as $value){                //遍历子节点
                         ?>
                         <tr>
                       <td width ="10%"   align="right" >
                               <img   src="images/childnode.gif" width="15" height="15" border="0" > </img>
                         </td>
                         <td   width ="90%"   align="left"   ><a href="#" >  <?php echo $value?></a>
                         </td>
                         </tr>
                   <?php
```

```
                    }
                 ?>
              </table>
           </div>
        </td>
      </tr>
    <?php
    }
       ?>
    </table>
  </td>
 </tr>
</table>
<script language="javascript">
function ShowNode(ParentID) {                                    //隐藏或显示子节点
      var Div = ParentID;
      if (Div.style.display == "none"){
           Div.style.display = "block";                           //显示 DIV，以此来显示子节点
      }else if (Div.style.display == "block"){
           Div.style.display = "none";                            //隐藏 DIV，间接隐藏子节点
      }
}
</script>
</form>
```

（3）人事组织的架构存储于 XML 文件 org.xml 中。

秘笈心法

心法领悟 216：动态设置超链接。

在设计网站项目时，有时需要根据实际情况动态设置超链接的内容。在利用<a>标签定义超链接时可以结合 PHP 脚本来实现这一功能，例如：

```
<a href="#" onclick= "ShowNode(<?php echo $node ?>)">   <?php echo $node ?></a>
```

实例 217	收缩式导航菜单 光盘位置：光盘\MR\03\217	高级 趣味指数：★★★★

实例说明

在网站中不仅可以设置导航条，而且还可以设置导航菜单。由于菜单内容比较多，在同一页面中显示会比较杂乱，所以目前多数设计者都采用收缩式的导航菜单。运行本实例，当浏览者单击“网站管理”时，在其下方将弹出导航菜单，如图 3.73 所示，浏览者再次单击“网站管理”时，导航菜单又收缩回去，页面中不再显示菜单中的内容。



图 3.73　收缩式菜单

关键技术

本实例主要是利用显示、隐藏表格来实现收缩式导航菜单的功能。单击导航菜单，显示当前菜单的内容，隐藏上一个显示的菜单。在隐藏菜单时，让其有规律地隐藏进而实现动画效果。



设计过程

（1）显示菜单的自定义函数，代码如下：

```
<SCRIPT language=javascript>
function show(obj,maxg,obj2){
  if(obj.style.pixelHeight<maxg) {
    obj.style.pixelHeight+=maxg/10;
obj.filters.alpha.opacity+=20;
obj2.background="images/title_hide.gif";
    if(obj.style.pixelHeight==maxg/10)
  obj.style.display='block';
myObj=obj;
mymaxg=maxg;
myObj2=obj2;
setTimeout('show(myObj,mymaxg,myObj2)','5');
  }
}
</SCRIPT>
```

（2）隐藏菜单的自定义函数，代码如下：

```
<SCRIPT language=javascript>
function hide(obj,maxg,obj2){
  if(obj.style.pixelHeight>0) {
    if(obj.style.pixelHeight==maxg/5)
  obj.style.display='none';
    obj.style.pixelHeight-=maxg/5;
obj.filters.alpha.opacity-=10;
obj2.background="images/title_show.gif";
myObj=obj;
mymaxg=maxg
myObj2=obj2;
setTimeout('hide(myObj,mymaxg,myObj2)','5');
  }
  else
    if(whichContinue)
  whichContinue.click();
}
</SCRIPT>
```

（3）单击菜单上的文字时，隐藏前一个菜单，显示当前菜单，代码如下：

```
<SCRIPT language=javascript>
function chang(obj,maxg,obj2){
  if(obj.style.pixelHeight) {
    hide(obj,maxg,obj2);
    nopen='';
    whichcontinue='';
  }
  else
    if(nopen){
      whichContinue=obj2;
      nopen.click();
}
else{
  show(obj,maxg,obj2);
  nopen=obj2;
  whichContinue='';
}
}
</SCRIPT>
```

（4）在表格的相关鼠标事件中调用自定义的方法和属性来改变收缩菜单的显示和隐藏，代码如下：

```
<TD class=list_title id=list1   onmouseover="this.typename='list_title2';"   onclick=chang(menu1,60,list1);   onmouseout="this.typename='list_title';"
background="images/title_hide.gif"   height=25><SPAN>网站管理</SPAN> </TD>
```

秘笈心法

心法领悟 217：收缩式导航菜单。

收缩式导航菜单是笔者最喜欢的导航菜单，虽然只是由一些 JavaScript 事件组成，但是操作时的动态效果很强。

实例 218 展开式导航菜单

光盘位置：光盘\MR\03\218

高级

趣味指数：★★★★

实例说明

对于一个企业网站来说，“联系我们”导航条是必不可少的。为了更加引人注意，可以将其制作成具有展开式动画效果的导航条。运行本实例，可以看到页面右侧的“联系我们”导航条是逐渐展开的，并且每次刷新页面时，都会以同样的动画效果展开，展开效果如图 3.74 所示。



图 3.74　展开式导航条

关键技术

本实例主要是通过自定义 JavaScript 函数控制图片（image）对象的 height 属性值实现的。为了实现动画效果，在 JavaScript 函数中还需要应用 setTimeout()方法延迟执行改变图片高度的操作。

设计过程

（1）在网页中显示“联系我们”导航条的位置插入一张全部展开后的“联系我们”导航图片，在需要设置超链接的文字上添加图片热点并设置相应的超链接。

（2）将图片的 height 属性值设置为 0，name 属性设置为 our。

（3）编写自定义的 JavaScript 函数 ourmove()，用于实现图片的展开效果。ourmove()函数的代码如下：

```
<script language="javascript">
function ourmove(){
 if(our.height<163){
        our.height=our.height+3
        setTimeout(ourmove,1)
 }
}
</script>
```

（4）在当前页面的最底部也就是</body>标记之前，添加如下代码实现导航条的展开式动画效果。

```
<script language="javascript">
ourmove();
</script>
```

秘笈心法

心法领悟 218：使用 ourmove()函数的注意事项。

一定要将调用 ourmove()函数的代码放在</body>标记之前，否则页面的其他内容需要等动画显示完毕才可以显示，达不到预期的效果。



实例 219　解释型菜单

光盘位置：光盘\MR\03\219

高级

趣味指数：★★★★

实例说明

在设计网站的导航菜单时，有时为了让用户快速了解菜单导航的详细内容，需要为菜单设置提示信息。这样当用户鼠标移动到菜单项上时就会显示提示信息，效果如图 3.75 所示。

明日图书　编程社区　企业简介　意见反馈

为图书用户提供在线答疑服务

图 3.75　解释型菜单

关键技术

本实例主要应用超链接中的 title 属性实现解释型菜单。在定义超链接时为其设置 title 属性，就可以实现菜单的提示功能，例如：

```
<a href="#" title = "为图书用户提供在线答疑服务">  明日图书   </a>
```

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 3 行 1 列的表格，在表格的第 2 行再添加一个 1 行 4 列的表格。

（3）在嵌套的表格中加入超链接，主要代码如下：

```
<td width="123" scope="col" align="right"> <a href="#" title = "为图书用户提供在线答疑服务">  明日图书   </a></td>
<td width="79" scope="col" align="right"><a href="#" title = "为编程人员提供一个在线交流平台">编程社区  </a></td>
<td width="78" scope="col" align="right"><a href="#">企业简介  </a></td>
<td width="82" scope="col" align="right"><a href="#">意见反馈  </a></td>
```

秘笈心法

心法领悟 219：使用 CSS 样式统一超链接效果。

在设计表单页面时，通常需要统一某一类表单元素的效果。例如，统一超链接文本字体的大小、文本颜色等。一种简单的方式是利用 CSS 样式来实现，例如：

```
<style>
a{
font-size:16px;
color:#900;
}
```

这样页面中的超链接字体均为 16 个像素，颜色为暗红色。

实例 220　半透明背景的下拉菜单

光盘位置：光盘\MR\03\220

高级

趣味指数：★★★★

实例说明

网站中的导航菜单是网站首页必不可少的表单元素，美观、新颖的导航菜单会为网站增色许多。本实例将实现一个半透明效果的菜单，效果如图 3.76 所示。



图 3.76　半透明效果菜单

关键技术

半透明效果的菜单可以通过 CSS 样式中的滤镜 filter 来实现。在滤镜中使用 alpha 属性设置透明程度（范围为 0~100，0 表示完全透明，100 表示不透明），其语法格式如下：

```
{Filter:alpha(opacity=opacity,finishopacity=finishopacity,sytle=sytle,startx=startx,starty=starty,finishx=finishx,
finishx=finishyx)}
```

例如，下面的语句将滤镜透明度设置为 50%。

```
Filter: Alpha(Opacity=50);
```

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 3 行 1 列的表格，在第 2 行中再添加一个 1 行 3 列的表格。

（3）定义 CSS 样式表，设置超链接的效果和半透明菜单的效果，代码如下：

```
<style type="text/css">
A {
COLOR: blue; FONT-SIZE: 10pt; TEXT-DECORATION: none;color:#FF6600
}
A:hover {
COLOR: black; FONT-SIZE: 10pt;color:#395531
}
.transMenu {
        BORDER: #666666 1px solid;
        FONT: 12px Verdana;
        POSITION: absolute;
        background-image:url("images/item_out.jpg");
        background-repeat : repeat-y;
        Filter: Alpha(Opacity=50);
}
</style>
```

（4）利用 JavaScript 脚本控制菜单的显示与隐藏，代码如下：

```
<script language="javascript">
function ShowMenu(divID, menuItem){                              //显示子菜单
        if (document.all){
                switch (menuItem){
                        case '网站导航':
                        {
                                var htmltxt = "<a href = 'www.mingribook.com'> 明日图书网站 </a> <br> <a href =
 'www.mrbccd.cn'> 明日编程社区 </a> <br> <a href = 'www.mrsoft.com'> 明日软件 </a> <br>";
                                MenuID.innerHTML = htmltxt;              //设置 DIV 代码
                                MenuID.style.padding = "5px";            //设置 DIV 边距
                                MenuID.style.height = "55px";            //设置 DIV 高度
                                break;
                        }
                        case '技术支持':
                        {
                                var menutxt = "<a href = 'www.mingribook.com'> QQ </a> <br> <a href = 'www.mrbccd.cn'>
 MSN </a> <br> <a href = 'www.mrsoft.com'> Phone </a> <br>";
                                MenuID.innerHTML = menutxt;              //设置 DIV 代码
                                MenuID.style.padding = "5px";            //设置 DIV 边距
                                MenuID.style.height = "55px";            //设置 DIV 高度
```

```
                                break;
                        }
                }
                //设置菜单显示位置
                MenuID.style.left = document.body.scrollLeft + event.clientX - event.offsetX;
                MenuID.style.top = document.body.scrollTop + event.clientY - event.offsetY + 18 ;
                MenuID.style.display = "block";
        }
}
function HideMenu(){                                                    //隐藏子菜单
        if (document.all){
                MenuID.style.display = "none";                          //隐藏 DIV
        }
}
if (document.all){
        document.onmousemove = HideMenu;                                //鼠标在表单中移动时隐藏菜单
}
</script>
```

秘笈心法

心法领悟 220：设置鼠标移动到超链接的效果。

在使用超链接时通常会为超链接设置一些特效。例如，当用户鼠标移动或单击超链接时文本颜色发生变化。要实现这一功能最简单的方式是定义 CSS 样式表，例如：

```
A:hover {
FONT-SIZE: 10pt;
color:#395531
}
```

实例 221　二级导航菜单

光盘位置：光盘\MR\03\221

高级

趣味指数：★★★★

实例说明

二级导航菜单是许多网站通常采用的菜单导航形式。顶行显示一级菜单，其下显示二级菜单。当用户选择不同的一级菜单时，二级菜单会动态发生变化。本实例就实现这种效果的导航菜单，如图 3.77 所示。

论坛中心 | 勘误发布 | 作品展示 | 联系我们

Visual C++ | Visual Basic | PHP

图 3.77　二级导航菜单

关键技术

实现二级导航菜单的关键是二级菜单的显示。当用户选择不同的顶层菜单时需要显示不同的二级菜单。这里我们用 DIV 作为二级导航菜单的容器，通过动态设置 DIV 对象的 innerHTML 属性来实现动态显示二级菜单的效果。

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 2 行 1 列的表格，在第 1 行中再添加一个 1 行 4 列的表格，在第 2 行再添加一个 1 行 2 列的表格。表单代码如下：

```
<form id="form1" name="form1" method="post" action="">
  <table width="610" height="441" border="0" align="center">
    <tr>
```

```
        <th height="62" align="right" valign="bottom" scope="col"><table width="337" border="1" align="right" cellpadding="0" cellspacing="0"
bordercolor="#993399" background="images/menuBK.JPG">
            <tr>
                <td width="327" scope="col"> <a href ='#' onmousemove= MenuCtrl('论坛中心','subMenu');>论坛中心</a></td>
                <td width="327" scope="col"><a href = '#' onmousemove=MenuCtrl('勘误发布','subMenu');>勘误发布</a></td>
                <td width="327" scope="col"><a href = '#' onmousemove=MenuCtrl('作品展示','subMenu');>作品展示</a></td>
                <td width="327" scope="col"><a href = '#' onmousemove=MenuCtrl('联系我们','subMenu');>联系我们</a></td>
            </tr>
        </table></th>
      </tr>
      <tr valign="top">
        <td height="373"><table width="606" border="0" align="center">
            <tr>
                <th width="283" scope="col"> </th>
                <th width="313" scope="col"><div id="subMenu" align="left"></div></th>
            </tr>
        </table></td>
      </tr>
    </table>
</form>
```

（3）利用 JavaScript 脚本控制二级菜单的显示，代码如下：

```
<script language="javascript">
 var curSel = "";
 function   MenuCtrl(menuID, divID){
        var divObj = document.getElementById(divID);                            //获取 DIV 对象
        if (divObj == null){
                return;
        }
        switch (menuID){                                                        //判断顶层菜单
                case '论坛中心':
                {
                        //在 DIV 中显示子菜单
                        divObj.innerHTML = "<a href = '#'>Visual C++ |<a href = '#'> Visual Basic |<a href = '#'> PHP </a>";
                        curSel ='论坛中心';
                        break;
                }
                case '勘误发布':
                {
                        //在 DIV 中显示子菜单
                        divObj.innerHTML = "<a href = '#'>光盘错误  |<a href = '#'>  文档错误  |<a href = '#'>目录错误  </a>";
                        curSel = '勘误发布';
                        break;
                }
                case '作品展示':
                {
                        //在 DIV 中显示子菜单
                        divObj.innerHTML = "<a href = '#'>视频学 VB |<a href = '#'>  视频学 VC |<a href = '#'>
视频学 Asp.net |<a href='#'>  视频学 C# </a>";
                        curSel = '作品展心';
                        break;
                }
                case '联系我们':
                {
                        //在 DIV 中显示子菜单
divObj.innerHTML = "<a href = '#'>OICQ |<a href = '#'> MSN|<a href = '#'> Phone</a>";
                        curSel ='联系我们';
                        break;
                }
        }
}
if (document.all){
        MenuCtrl('论坛中心', 'subMenu');                                        //默认情况下显示第一个顶层菜单的二级菜单项
}
</script>
```

秘笈心法

心法领悟 221：设置表格单边框效果。

默认情况下，表格具有双边框，在利用表格进行界面布局时这种效果会影响页面的美观性，可以通过将表格的 cellspacing 属性设置为 0 来实现表格的单边框效果，例如：

```
<table width="337" border="1"   cellspacing="0" bordercolor="#993399" background="images/menuBK.JPG">
```

实例 222	导航条的动画效果 光盘位置：光盘\MR\03\222	高级 趣味指数：★★★★

实例说明

导航条是网站设计中不可缺少的元素之一，它能正确地引导浏览者查找需要的资料，成为浏览者的网站路标。同时网页导航条的设计风格也能影响到页面的整体风格，对于一个静态元素居多的页面，可以为导航条添加动态效果，这样可以使整个网站不至于太呆板，增强网页的欣赏价值。例如，在明日实业网站中，笔者将导航条设计成动画效果，用户将鼠标移动到任意一个导航按钮上时，该按钮都会突出显示，鼠标移出后，又恢复为原来的位置，运行结果如图 3.78 所示。



图 3.78　导航条的动画效果

关键技术

本实例主要是通过 Image 对象的鼠标事件控制 Image 对象的 src 属性的值实现的。

在 JavaScript 里提供了对图像进行处理的专用对象 Image 来装入文档的图形。Image 对象与其他对象的差别在于，其允许通过构造器显示和创建新的 Image 对象，创造和预装入的图形之前并非 Web 页面的组成部分。Image 对象存在于浏览器的缓冲区中，用于替换已经显示的图像。

用 Image 构造器创建图形的语法格式如下：

```
ObjImg=new image()                              //创建了一个新的 Image 对象，并将其赋予变量 ObjImg
ObjImg.src="图片文件相对路径"                    //设置 Image 对象的 Src 属性
```

设计过程

（1）准备 14 张图片。7 张鼠标移出时显示的图片，图片的名称为“menu_0+1~7 的数字+.gif”，7 张鼠标移入时显示的图片，图片的名称为“menu_0+1~7 的数字+_over.gif”。

（2）将准备的 7 张鼠标移出时显示的图片按顺序插入页面中的适当位置，并设置其鼠标事件 onMouseMove 和 onMouseout 执行的操作，这里分别调用两个不同的自定义 JavaScript 函数，关键代码如下：

```
<img src="Images/top/menu_01.gif" id="image1" width="95" height="119" border="0" onMouseMove="move(this,'1')" onMouseout="out(this,'1')">
<img src="Images/top/menu_02.gif" id="image2" width="95" height="119" border="0" onMouseMove="move(this,'2')" onMouseout="out(this,'2')">
<img src="Images/top/menu_03.gif" name="image3"   width="95" height="119" border="0" onMouseMove="move(this,'3')"
onMouseout="out(this,'3')">
<img src="Images/top/menu_04.gif" name="image4" width="94" height="119" border="0" onMouseMove="move(this,'4')" onMouseout="out(this,'4')">
<img src="Images/top/menu_05.gif" name="image5" width="95" height="119" border="0" onMouseMove="move(this,'5')" onMouseout="out(this,'5')">
```

```
<img src="Images/top/menu_06.gif" name="image6" width="94" height="119" border="0" onMouseMove="move(this,'6')" onMouseout="out(this,'6')">
<img src="Images/top/menu_07.gif" width="95" name="image7" height="119" border="0" onMouseMove="move(this,'7')" onMouseout="out(this,'7')">
```

（3）编写自定义的 JavaScript 函数 move()和 out()，move()函数用于设置鼠标移入导航按钮时显示的图片，out()函数用于设置鼠标移出导航按钮时显示的图片，代码如下：

```
<script language="javascript">
//鼠标移动效果
var A_Img=new Image();
function move(image,num){
 image.src='Images/top/menu_0'+num+'_over.gif';
}
function out(image,num){
image.src='Images/top/menu_0'+num+'.gif';
}
</script>
```

秘笈心法

心法领悟 222：Image 构造器。

在利用 Image 对象编程之前，利用<img>标记显示属性。在 HTML 中，要显示图片可以用如下语句来实现：

```
<img src="pictureName.gif" name="ImgName">
```

在 JavaScript 中可以用下面的方式访问 Image 对象：

```
document.Img.src=" menu_01.gif";
```

第 2 篇

常用技术篇

第 4 章

文件操作

- 文件上传
- 文件下载
- 文件遍历
- 文件操作
- 文件的压缩与解压
- 目录操作

4.1　文 件 上 传

文件上传是网站中一个非常重要的功能，特别是在网站的后台管理系统中，需要通过此功能向网站中上传很多数据，包括图片、文件等。

实例 223　以二进制形式上传文件到数据库

光盘位置：光盘\MR\04\223

中级　趣味指数：★★☆

实例说明

文件的上传可以分为两类：一类是上传文件到服务器，即存储到服务器指定的文件夹下；另一类是将数据以二进制的形式上传到数据库中。具体采用哪种方式，需要根据程序的实际需要而定。

本实例中将介绍如何将文件以二进制的形式上传到数据库中，并且输出上传的数据。运行结果如图 4.1 所示。



图 4.1　将文件以二进制的形式上传到数据库

关键技术

本实例主要应用表单中的文件域提交图片文件，代码如下：

```
<input name="cover" type="file" id="cover" size="30">
```

其中，参数 type 指定表单元素的类型为文件域；name 指定表单元素的名称；size 指定表单元素的大小。

通过$_POST 方法获取表单中提交的数据，并应用 insert 语句将表单中的数据添加到数据表中。insert 语句的语法如下：

```
Insert into table_name (column_name1,column_name2, … ) values (value1, value2, … )
```

参数说明：

table_name：数据表的名称。

column_name1：数据表中字段的名称。

valuel：表单提交的变量。

在 index_ok.php 文件中，通过 fopen()函数和 fread()函数读取表单中提交的图片数据，并且将返回的数据存储到数据库中。

在这里要特别注意网页编码格式的使用。如果网页的编码格式是 UTF-8，那么在获取上传文件的存储路径时，必须将 UTF-8 编码格式的字符串转换为 GB2312 编码格式，否则文件不能被 fopen()函数读取。

设计过程

（1）通过 Dreamweaver 开发工具创建一个 index.php 页，添加一个表单，设置表单的 method 属性值为 post，Action 属性值为 index_ok.php；添加表单元素，提交图书信息；添加提交按钮，其关键代码如下：

```
<form name="form" method="post" action="index_ok.php" onSubmit="return check_form(this)">
<input name="cover" type="file" id="cover" size="30">
<input type="submit" name="Submit" value="提交">
</form>
```

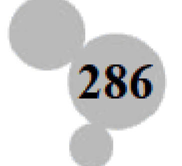
（2）创建 conn 文件夹，编写 conn.php 文件，连接 MySQL 数据库服务器，连接 db_database04 数据库，设置数据库变量格式为 utf8。

（3）创建 index_ok.php 文件，通过$_POST 方法获取表单中提交的数据，并判断上传图片的格式是否正确，通过 fopen()函数和 fread()函数读取表单中提交的图片数据，编写 insert 语句将数据添加到指定的数据表中，其关键代码如下：

```
<?php
include ("conn/conn.php");                                              //连接数据库
$bookname = $_POST["bookname"];                                         //获取表单中提交的数据
$price = $_POST["price"];
$maker = $_POST["maker"];
$issuDate = date("Y-m-d H:i:s");
$publisher = $_POST["publisher"];
$synopsis = $_POST["synopsis"];
if ($_POST["Submit"] == true) {
    $cover = $_FILES["cover"]['name'];                                  //获取表单中提交的图片
    $cover_type = strstr($cover, ".");                                  //获取从“.”到最后的字符
    if ($cover_type != ".jpg" && $cover_type != ".gif" && $cover_type !=
     ".JPG" && $cover_type != ".GIF" && $cover_type != ".bmp" && $cover_type !=
     ".BMP") {                                                          //判断图片的格式
        echo "<script>alert('封面图片格式不对，请进行处理后在上传！'); window.location.href='index.php';</script>";
    } else {
        $cover=iconv("utf-8","gb2312",$cover);                         //设置字符串的编码格式
        $path = "uploadfiles/".$cover;
        @move_uploaded_file($_FILES["cover"]["tmp_name"],$path);
        $fp = fopen($path, "rb");                                       //以二进制形式打开图片
        $image = addslashes(
        @fread($fp, filesize($path)));                                  //读取二进制的数据
        //将数据添加到指定的数据表中
        $sql = "insert into
tb_book(bookname,price,maker,issuDate,publisher,synopsis,cover)values('$bookname','$price','$maker','$issuDate','$publisher','$synopsis','$image')";
        $result = mysql_query($sql,
        $conn);
        if ($result == true) {
            echo iconv("utf-8","gb2312","文件上传成功!!");
            echo "<meta http-equiv=\"refresh\" content=\"30 url=index.php\">";
        } else {
            echo iconv("utf-8","gb2312","文件上传失败!!");
            echo "<meta http-equiv=\"refresh\" content=\"30 url=index.php\">";
        }
    }
}
?>
```

秘笈心法

心法领悟 223：读取数据库中存储的二进制图片数据。



读取数据库中存储的二进制图片数据需要定义一个单独的文件，根据数据的 ID 值执行查询，并应用 mysql_reult()函数返回查询结果，最后输出查询结果。本实例中定义的是 images.php 文件，通过该文件完成对数据库中存储的二进制图片的读取操作。images.php 的代码如下：

```
<?php
include ("conn/conn.php");                                     //连接数据库
$query = "select * from tb_book where id=" . $_GET[recid];     //定义查询指定数据的语句
$result = mysql_query($query);                                 //执行查询
if (! $result)
    die("error: mysql query");                                 //返回失败的结果
$num = mysql_num_rows($result);                                //获取查询结果的数量
if ($num < 1)
    die("error: no this recorder");                            //如果查询结果小于则说明没有数据
$data = mysql_result($result, 0, "cover");                     //获取查询结果
mysql_close($conn);                                            //关闭数据库
echo $data;                                                    //输出返回的数据
?>
```

实例 224　上传文件到服务器　高级

光盘位置：光盘\MR\04\224

趣味指数：★★★★☆

实例说明

在开发网站的过程中，不但可以将文件上传到数据库中，而且可以将文件上传到服务器，这种方式可以为网站中的数据库节省很多空间，同时可以控制上传文件的大小、格式，并且读取服务器中的文件要比从数据库中读取方便很多。运行本实例，如图 4.2 所示，单击图中的“浏览”按钮，选择要上传的文件，然后单击“提交”按钮，如果上传成功，则在页面中提示“上传成功！”，否则给出错误提示信息。



图 4.2　上传文件

关键技术

在 PHP 中，应用 move_uploaded_file()函数实现将文件上传到指定文件夹。但是，在执行文件上传之前，为了防止潜在的攻击对原本不能通过脚本交互的文件进行非法管理，可以先应用 is_uploaded_file()函数判断指定的文件是否通过 HTTP POST 上传，如果是则返回 TRUE。is_uploaded_file()函数的语法如下：

```
bool is_uploaded_file ( string filename )
```

参数 filename 必须指定类似于$_FILES['filename']['tmp_name']的变量，不可以使用从客户端上传的文件名$_FILES['filename']['name']。

通过 is_uploaded_file()函数对上传文件进行判断，可以确保恶意的用户无法欺骗脚本去访问本不能访问的文件，如/etc/passwd。

move_uploaded_file()函数将文件上传到服务器中指定的位置。如果成功返回 TRUE，否则返回 FALSE，其

语法如下：

```
bool move_uploaded_file ( string filename, string destination )
```

参数 filename 指定上传文件的临时文件名，即$_FILES[tmp_name]；参数 destination 指定文件上传后保存的新路径和名称。

说明：如果参数 filename 不是合法的上传文件，不会出现任何操作，move_uploaded_file()函数将返回 FALSE。如果参数 filename 是合法的上传文件，但出于某些原因无法移动，也不会出现任何操作，move_uploaded_file()函数将返回 FALSE，并且还会发出一条警告。

设计过程

（1）创建 index.php 文件，添加表单，设置 enctype 属性值为"multipart/form-data"，添加文件域、提交按钮，使用 POST 方法，将表单中的数据提交到本页。

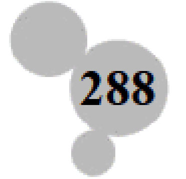
（2）通过$_FILES 获取上传文件的相关信息，通过 move_upload_file()函数完成图片的上传，代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                                    //判断上传内容是否为空
 if($_FILES['up_picture']['error']>0){                                    //判断文件是否可以上传到服务器
      echo "上传错误:";
      switch($_FILES['up_picture']['error']){
           case 1:
                echo "上传文件大小超出配置文件规定值";
           break;
           case 2:
                echo "上传文件大小超出表单中约定值";
           break;
           case 3:
                echo "上传文件不全";
           break;
           case 4:
                echo "没有上传文件";
           break;
      }
 }else{
      if(!is_dir("./upfile/")){                                           //判断指定目录是否存在
           mkdir("./upfile/");                                            //创建目录
      }
      $path='./upfile/'.time().strstr($_FILES['up_picture']['name'],'.');  //定义文件名称和存储位置
      if(is_uploaded_file($_FILES['up_picture']['tmp_name'])){            //是否通过 HTTP POST 上传
           if(!move_uploaded_file($_FILES['up_picture']['tmp_name'],$path)){   //执行上传
                echo "上传失败";
           }else{
                echo "文件".time().$_FILES['up_picture']['name']."上传成功，大小为：".$_FILES['up_picture']['size'];
           }
      }else{
           echo "上传文件".$_FILES['up_pictute']['name']."不合法！";
      }
 }
}
?>
```

秘笈心法

心法领悟 224：通过表单中的隐藏域控制上传文件的大小。

如果应用$_FILES['picture']['error']对上传文件的信息进行判断，当其返回值为 2 时，说明上传文件的大小超出了表单中约定的范围，而表单中的这个约定值是通过隐藏域 MAX_FILE_SIZE 的值来控制的。在设置这个隐藏域时，必须将其放置在文件域之前，否则不起作用。



实例 225　限制上传文件的大小

光盘位置：光盘\MR\04\225

高级

趣味指数：★★★★☆

实例说明

在网站开发的过程中，为了确保充分地利用服务器的空间，在开发上传功能时最好能够对上传文件的大小进行控制。本实例开发的上传功能就可以对上传文件的大小进行控制。如果上传文件的大小超过指定范围，那么将给出提示信息，并终止上传。运行效果如图 4.3 所示。



图 4.3　限制上传文件的大小

关键技术

控制上传文件的大小有两个关键点：一是 PHP 的配置文件 php.ini 对上传文件的控制，如果上传文件的大小超过其指定的范围，那么上传就会失败；二是在 PHP 配置文件允许的范围内，在程序中对上传文件大小的控制。

（1）在 PHP 中，通过 php.ini 文件对上传文件进行控制，包括是否支持上传、上传文件的临时目录、上传文件的大小、指令执行的时间、指令分配的内存空间。

在 php.ini 文件中，定义到 File Uploads 项，完成对上传相关选项的设置。上传相关选项的含义如下：

☑ file_uploads：如果值是 on，说明服务器支持文件上传；如果为 off，则不支持。一般默认是支持的，不需修改。

☑ upload_tmp_dir：上传文件的临时目录。在文件被成功上传之前，首先存放到服务器端的临时目录中。多数使用系统默认目录，但是也可以自行设置。

☑ upload_max_filesize：服务器允许上传文件的最大值，以 MB 为单位。系统默认为 2MB，如果网站需要上传超过 2MB 的数据，那么就要修改这个值。

上述是 php.ini 文件的 File_Uploads 项中与上传相关选项参数的设置说明，除了 File_Uploads 项中的内容外，在 php.ini 文件中还有几个选项会影响到文件的上传。

☑ max_execution_time：PHP 中一个指令所能执行的最大时间，单位是秒。在上传超大文件时必须修改该选项，否则即使上传文件在服务器允许的范围内，若是超过了指令所能执行的最大时间，仍然无法实现上传。

☑ memory_limit：PHP 中一个指令所分配的内存空间，单位是 MB，其大小同样会影响到超大文件的上传。

（2）在客户端控制上传文件，应用的是 form 表单中的 enctype 和 method 属性，以及隐藏域 MAX_FILE_SIZE。

☑ enctype="multipart/form-data"：指定表单编码数据方式。

☑ method="post"：指定数据的传输方式。

☑ <input type="hidden" name="MAX_FILE_SIZE" value="10000" />：通过隐藏域控制上传文件的大小（单位为字节），该值不能超过 php.ini 配置文件中 upload_max_filesize 选项设置的值。它不能完全控制上传文件的大小，只是可以避免一些不必要的麻烦。

上述两种判断的结果都可以通过全局变量$_FILES 的返回值来体现。$_FILES 是一个数组，包含所有上传文件的信息。下面介绍$_FILES 数组中每个元素的含义，如表 4.1 所示。

表 4.1　$_FILES 数组中的元素

元　素　名	说　　明
$_FILES[filename][name]	存储上传文件的文件名，如 text.txt、title.jpg 等
$_FILES[filename][size]	存储文件大小，单位为字节
$_FILES[filename][tmp_name]	存储文件在临时目录中使用的文件名。因为文件在上传时，首先要将其以临时文件的身份保存在临时目录中
$_FILES[filename][type]	存储上传文件的 MIME 类型，MIME 类型规定各种文件格式的类型。每种 MIME 类型都是由“/”分隔的主类型和子类型组成，如 image/gif，主类型为“图像”，子类型为 GIF 格式的文件；text/html 代表文本的 HTML 文件
$_FILES[filename][error]	存储与文件上传相关的错误代码。此项目是 PHP 4.2.0 版本中新增的内容，其返回值有以下 5 种： 0：表示没有任何错误，文件上传成功 1：表示上传文件的大小超出了 PHP 配置文件指令 upload_max_filesize 选项限制的值 2：表示上传文件的大小超出了 HTML 表单中 MAX_FILE_SIZE 选项所指定的值 3：表示文件只被上传了一部分 4：表示没有上传任何文件

设计过程

（1）创建 index.php 文件。

（2）为 index.php 文件添加表单，设置 enctype 属性值为"multipart/form-data"，添加文件域、提交按钮，使用 POST 方法将表单中数据提交到本页。

（3）通过$_FILES 获取上传文件的相关信息，通过 move_upload_file()函数完成图片的上传，代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                                  //判断上传内容是否为空
 if($_FILES['up_picture']['error']>0){                                  //判断文件是否可以上传到服务器
        echo "上传错误:";
        switch($_FILES['up_picture']['error']){
                case 1:
                        echo "上传文件大小超出配置文件规定值";
                break;
                case 2:
                        echo "上传文件大小超出表单中约定值";
                break;
                case 3:
                        echo "上传文件不全";
                break;
                case 4:
                        echo "没有上传文件";
                break;
        }
 }else{
        if(!is_dir("./upfile/")){                                       //判断指定目录是否存在
                mkdir("./upfile/");                                     //创建目录
```



```
		}
		$path='./upfile/'.time().strstr($_FILES['up_picture']['name'],'.');		//定义文件名称和存储位置
		if(is_uploaded_file($_FILES['up_picture']['tmp_name'])){			//是否通过 HTTP POST 上传
			if(!move_uploaded_file($_FILES['up_picture']['tmp_name'],$path)){	//执行上传
				echo "上传失败";
			}else{
				echo "文件".time().$_FILES['up_picture']['name']."上传成功，大小为：".$_FILES['up_picture']['size'];
			}
		}else{
			echo "上传文件".$_FILES['up_pictute']['name']."不合法！";
		}
	}
}
?>
```

秘笈心法

心法领悟 225：如何上传超大文件。

如果要上传超大文件，必须修改 php.ini 文件。修改选项包括：upload_max_filesize 的最大值、max_execution_time 一个指令所能执行的最大时间和 memory_limit 一个指令所分配的内存空间。

实例 226　限制上传文件的类型

光盘位置：光盘\MR\04\226

中级

趣味指数：★★★★☆

实例说明

在开发文件上传功能时，不仅要考虑上传文件的大小，有时还要考虑上传文件的类型，针对不同的需要对上传文件的类型加以限制，这里指的文件类型可以通过文件的扩展名（如.txt、.php、.doc）来判断。例如，在进行文件上传时，涉及文本说明性文字时，最好将上传的文件限制为以“.txt”为后缀的文本格式。

运行本实例，如图 4.4 所示，单击图中的“浏览”按钮，选择要上传的文件（文件不是以“.txt”为扩展名），单击“提交”按钮，程序将给出错误提示。



图 4.4　限制上传文件的类型

关键技术

文件的上传和对大小的限制已经在前面的实例中讲解过，这里只讲解限制上传文件类型的关键技术，首先要通过预定义全局变量$_FILES 中的$_FILES[files][name]获取上传文件的名称，然后应用 strstr()函数截取上传文件名称的后缀，最后判断上传文件的后缀是否符合要求。

strstr()函数的语法如下：

```
string strstr(string strings, string needle);
```

该函数用于在字符串 strings 中查找字符串 needle 的位置，并返回从 needle 字符串开始到 strings 字符串结束

的所有内容，如果在 strings 中没有查找到 needle，该函数将返回 FALSE。

设计过程

（1）继续使用实例 225 的内容，在其基础上增加对上传文件类型的判断，增加的关键代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                //判断上传内容是否为空
$type=$_FILES['up_picture']['name'];                  //获取上传文件的名称
$types=strstr($type,'.');                             //截取上传文件的后缀
if($types==".txt"){                                   //判断上传文件的后缀是否符合要求
        //省略了部分代码
}
}else{
                echo "上传文件".$_FILES['up_pictute']['name']."类型不正确！";
}
}
?>
```

（2）重新保存 index.php 文件。

秘笈心法

心法领悟 226：判断上传的文件是否符合多种类型中的一种。

判断上传的文件是否符合多种类型中的一种，最佳的方法就是将多种类型定义到一个数组中，然后通过 for 循环输出数组中的元素值，将输出的元素值与获取的文件后缀进行比较，进而判断其是否符合要求。

实例 227	上传多个文件到服务器 光盘位置：光盘\MR\04\227	高级 趣味指数：★★★★☆

实例说明

上传图片到服务器，是程序开发过程中必不可少的一个功能。它不但可以达到共享图片的目的，而且可以提高网站的访问量，丰富网站的内容。在本实例中将讲解如何通过 POST 方式实现多图片上传，运行结果如图 4.5 所示。



图 4.5　POST 方式实现多图片上传



关键技术

多文件上传的关键是如何定义上传文件元素的名称及判断上传文件的数量。在本实例中，以数组的形式定义上传文件的名称（上传文件的名称是“files[]”）。为了达到可以上传任意数量图片（4 个图片以内）的目的，在对上传文件进行处理的过程中应用 array_filter()函数和回调函数去除数组中的空元素。

array_filter()函数，用回调函数过滤数组中的单元，语法如下：

```
array array_filter ( array input [, callback callback] )
```

array_filter()函数依次将 input 数组中的每个值传递到 callback()函数。如果 callback()函数返回 TRUE，则 input 数组的当前值会被包含在返回的结果数组中，并且数组的键名保留不变。

说明： 在回调函数中不要对数组进行修改操作，例如，增加或者删除数组中的元素，如果一旦数组改变，那么此函数的运用也就没有意义了。如果没有提供 callback()函数，array_filter()函数将删除 input 中所有等值为 FALSE 的元素。

本实例中定义的回调函数是 check()，用于验证数组中的元素值是否为空，其语法如下：

```
function check($var) { //验证数组的返回值是否为空
        return ($var != "");
}
```

说明： 通过 POST 方法实现多图片上传，在创建 form 表单时，必须指定 enctype="multipart/form-data"属性。如果要通过隐藏域 MAX_FILE_SIZE 的值对上传文件的大小进行控制，那么必须将隐藏域放置在上传文件的文件域之前，否则是不会起作用的。

设计过程

（1）创建 index.php 文件。添加表单，设置文件域、提交按钮，使用 POST 方法设置 enctype="multipart/form-data"，将数据提交到 index_ok.php 页，完成多个文件的上传操作，其关键代码如下：

```
<table width="750" border="0" cellspacing="0" cellpadding="0">
    <form action="index_ok.html" method="post" enctype="multipart/form-data" name="form1">
        <tr>
          <td width="100" height="25" align="right" class="STYLE1">内容 1：</td>
          <td width="150" align="center"><input name="files[]" type="text" id="files[]" size="15"></td>
          <td align="left"><input name="picture[]" type="file" id="picture[]" size="30"></td>
          </tr>
        <tr>
          <td height="25" align="right" class="STYLE1">内容 2：</td>
          <td align="center"><input name="files[]" type="text" id="files[]" size="15"></td>
          <td align="left"><input name="picture[]" type="file" id="picture[]" size="30"></td>
          </tr>
        <tr>
          <td colspan="3" align="center">
              <input type="image" name="imageField" src="images/bg_09.jpg">    
              <input type="image" name="imageField2" src="images/bg_11.jpg"></td>
          </tr>
      </form>
    </table>
```

（2）在 index.php 文件中，连接数据库，读取数据库中存储的数据，实现上传文件的分页输出。代码请参考光盘中的相关内容。

（3）创建 index_ok.php 文件，获取表单中提交的数据，将多个文件存储到服务器中，将文件的名称和存储路径存储到数据库中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include "conn/conn.php";                                                //包含数据库链接文件
if ($_POST [files] != "") {
        if (! is_dir ( "./upfile" )) {
                mkdir ( "./upfile" );                                   //创建上传文件存储文件夹
```

```
        }
        $data = date ( "Y-m-d H:m:s" );                                         //定义时间
        function check($var) {                                                  //验证数组的返回值是否为空
                return ($var != "");                                            //如果不为空，则返回数组元素
        }
        $files = array_filter ( $_POST ["files"], "check" );                    //去除数组中的空值
        $array = array_filter ( $_FILES ["picture"] ["name"], "check" );        //去除数组中的空值
        foreach ( $array as $key => $value ) {                                  //循环读取数组中的数据
                $path = 'upfile/' . time () . $key . strtolower ( strstr ( $value, "." ) );   //定义上传文件存储位置
                move_uploaded_file ( $_FILES ["picture"] ["tmp_name"] [$key], $path );         //执行上传操作
                $query = "insert into tb_up_file (file_test,data,file_name) values ('$path','$data','$files[$key]')";
                $result = mysql_query ( $query );
        }
        echo "<script>alert('图片上传成功'); window.location.href='index.html';</script>";
}
?>
```

秘笈心法

心法领悟 227：通过伪静态技术隐藏 PHP 文件后缀。

首先，修改 Apache 服务器的配置文件 httpd.conf。打开 httpd.conf 文件，定位到如下位置：

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

将该项前面的“#”去掉后启动该项。

然后，查找 httpd.conf 文件，找到其中的 AllowOverride 项，将它的值都修改为 All。保存并重新启动 Apache 服务器，使修改生效。

最后，在实例根目录下创建.htaccess 文件，实现对 PHP 文件后缀的隐藏操作。.htaccess 文件的代码如下：

```
RewriteEngine On            #启动项
RewriteRule ^index.html$ index.php
RewriteRule ^index_ok.html$ index_ok.php
RewriteRule ^index-([0-9]+)-([0-9]+)-([0-9]+)\.html$ index.php?vv=$1&ljjl=$2&page=$3 [L]
```

通过正则表达式对文件的后缀和传递的参数进行匹配，完成对 PHP 文件后缀的隐藏操作。

4.2　文 件 下 载

实例 228	通过链接方式下载 光盘位置：光盘\MR\04\228	初级 趣味指数：★★☆

实例说明

在一个完整的网站中，具有文件下载功能能够给网站带来更多的访问者，增加网站的访问量。最常用的文件下载方式就是通过链接下载。在本实例中，应用链接方式进行下载，当单击“下载”超链接时，即可弹出文件“另存为”对话框，完成下载保存操作，运行结果如图 4.6 所示。

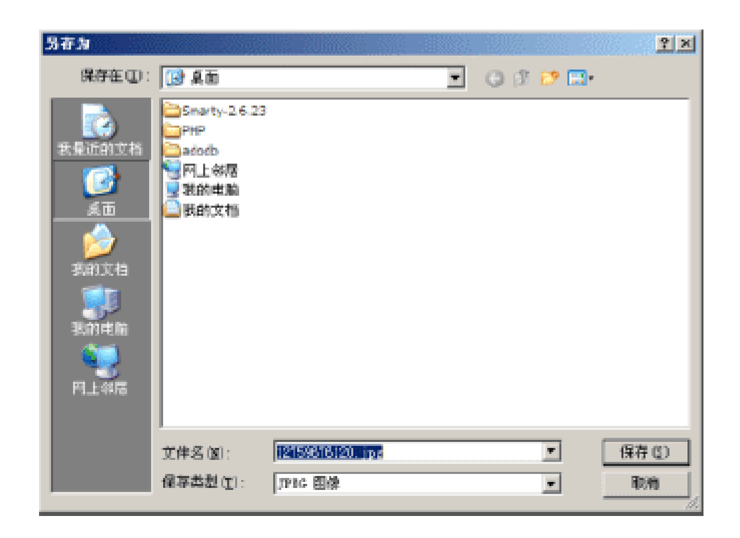


图 4.6　链接方式的文件下载



关键技术

本实例主要通过超链接来实现文件的下载。通过 select 语句从数据库中读取文件存储的路径，然后将该文件作一个超链接，当单击“下载”超链接时，即可下载该文件。

设计过程

（1）在实例 227 的基础上，为输出的图片名称创建超链接，完成图片的下载操作，其关键代码如下：

```
<a href="<?php echo $myrow[file_test];?>"><?php echo $myrow[file_name];?></a>
```

（2）保存对文件的修改，运行程序，运行结果参看实例说明。

秘笈心法

心法领悟 228：不同类型的文件，下载操作不同。

如果是压缩的文件，可以直接下载；如果不是，则需要右击，然后在弹出的快捷菜单中选择“另存为”命令。

实例 229	通过 header()函数进行下载 光盘位置：光盘\MR\04\229	中级 趣味指数：★★★★☆

实例说明

除了可以通过链接方式下载文件之外，还可以通过 header()函数完成下载操作。例如，本实例中就是应用 header()函数实现文件的下载，如果下载的文件不存在，则会给出提示信息，运行结果如图 4.7 所示。

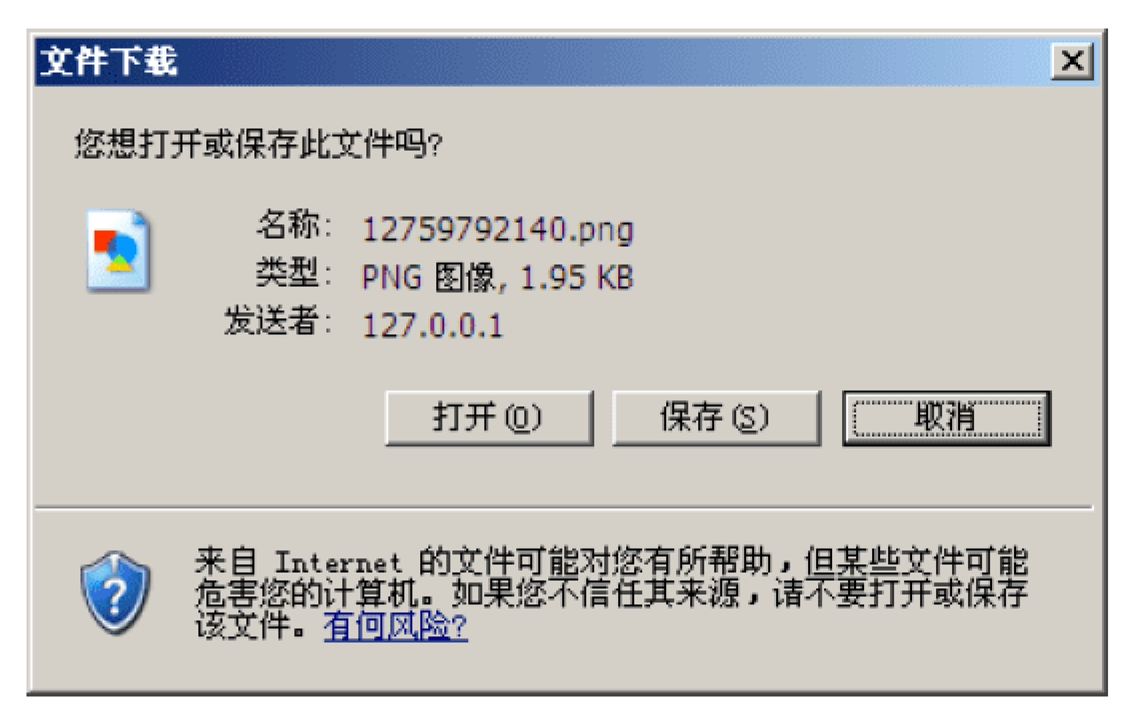




图 4.7　应用 header()函数实现文件下载

关键技术

通过 HTTP 方式下载文件，主要应用 header()函数。

header()函数属于 HTTP 函数，其作用是以 HTTP 协议将 HTML 文档的标头送到浏览器，并告诉浏览器具体怎么处理这个页面。header()函数的语法如下：

```
void header ( string string [, bool replace [, int http_response_code]] )
```

参数说明：

string：发送的标头。

replace：如果一次发送多个标头，对于相似的标头是替换还是添加。如果是 FALSE，则强制发送多个同类型的标头。默认是 TRUE，即替换。

http_response_code：强制 HTTP 响应为指定值。

通过 HTTP 下载的代码如下：

```
header("Content-type: application/x-gzip");
header("Content-Disposition: attachment; filename=文件名");
header("Content-Description: PHP3 Generated Data"); >
```

HTTP 标头有很多，这里介绍的是下载的 HTTP 标头，其代码如下：

```
header('Content-Disposition: attachment; filename="filename"');
```

在应用过程中，唯一需要改动的就是 filename，即将 filename 替换为要下载的文件。

设计过程

（1）在实例 227 的基础上，为输出的图片创建超链接，链接到 download.php 文件，通过 download.php 完成文件的下载操作。在 index.php 文件中所做的修改如下：

```
<td width="245" height="100" align="center" valign="middle" bgcolor="#F0F0F0"><a href="download-<?php echo $myrow[file_test];?>.html" title="
点击即可下载！ "><?php echo "<img src=\"$myrow[file_test]\" width=\"250\" height=\"100\" border=\"0\">"?></a></td>
```

（2）创建 download.php 文件，应用 header()函数实现文件的下载，其代码如下：

```
<?php
header("Content-type: text/html; charset=UTF-8");                    //设置文件编码格式
$path = $_GET['path'];                                               //获取文件路径
if(!empty($path) and !is_null($path)){                               //判断变量是否为空，是否为 NULL
        $filename=basename($path);                                   //获取文件名
        $file=@fopen($path,"r");
        if($file){
                header("Content-type:application/octet-stream");     //输出 MIME 类型
                header("Accept-ranges:bytes");                       //接受的范围单位
                header("Accept-length:".filesize($path));            //文件长度
                header("Content-Disposition:attachment;filename=".$filename);   //默认的文件保存对话框中的文件名称
                echo fread($file,filesize($path));                   //读取文件
                fclose($file);                                       //关闭文件
                exit;                                                //退出
        }else{
                echo "<script>alert('您下载的文件不存在！ '); history.back();</script>";
        }
}
?>
```

秘笈心法

心法领悟 229：header()函数的应用。

通过 header()函数不但可以实现文件的下载，还可以实现以下 3 个功能：

（1）重定向，这是最常用的功能。

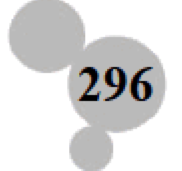
```
header("Location: http://www.mrbccd.com");
```

（2）强制客户端每次访问页面时获取最新资料，而不是使用存在于客户端的缓存。

```
//设置页面的过期时间（用格林威治时间表示）
header("Expires:     Mon,     08     Jul     2008     08:08:08     GMT");
//设置页面的最后更新日期（用格林威治时间表示），使浏览器获取最新资料
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "GMT");
header("Cache-Control: no-cache,     must-revalidate");       //控制页面不使用缓存
header("Pragma: no-cache");                                   //参数（与以前的服务器兼容），即兼容 HTTP 1.0 协议
header("Content-type:     application/file");                 //输出 MIME 类型
header("Content-Length:     227685");                         //文件长度
header("Accept-Ranges:     bytes");                           //接受的范围单位
//默认的文件保存对话框中的文件名称
header("Content－Disposition: attachment; filename=$filename");   //实现下载
```

（3）输出状态值到浏览器，控制访问权限。

```
header('HTTP/1.1 401 Unauthorized');
header('status: 401 Unauthorized');
```



4.3　文 件 遍 历

实例 230　读取整个文件的内容

光盘位置：光盘\MR\04\230

初级

趣味指数：★★☆

实例说明

在开发网站的过程中，很多的服务条款、协议等都是以文本文件的形式存储的。如果要读取这些文件中的内容，就需要应用到文件系统函数。在本实例中将介绍如何通过文件系统函数读取整个文件的内容，其运行效果如图 4.8 所示。

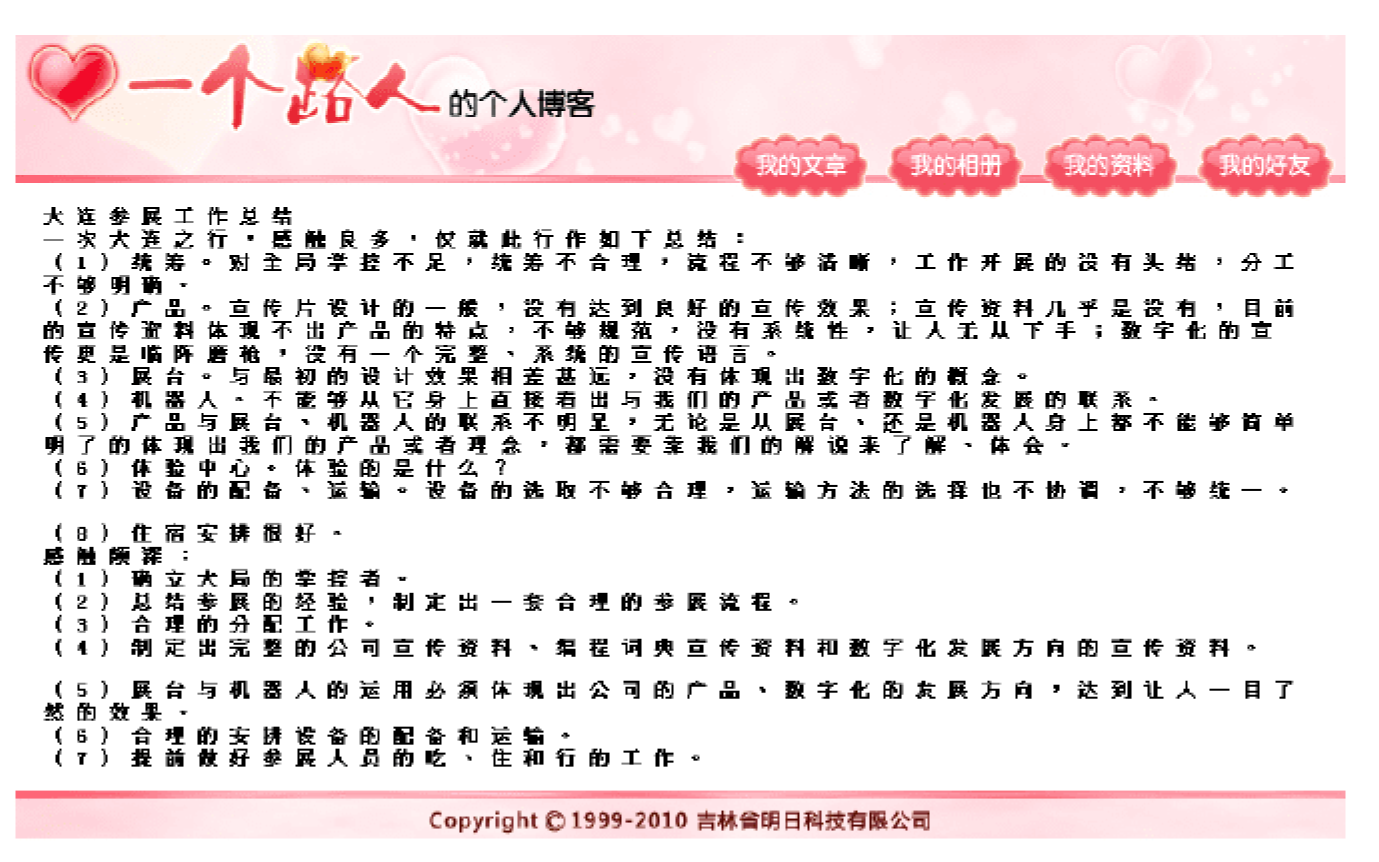


图 4.8　读取文本文件的内容

关键技术

在 PHP 中，读取整个文件的内容，有 3 个函数可以选择：

☑　readfile()函数

readfile()函数读取一个文件并将其写入输出缓冲，如成功，返回读取的字节数；如失败，返回 FALSE，语法如下：

```
int readfile ( string filename [, bool use_include_path [, resource context]] )
```

参数 filename 指定读取的文件名称；参数 use_include_path 控制是否支持在 include_path 中搜索文件，如果支持，则将该值设置为 TRUE；参数 context 是 PHP 5.0 的新增内容。

说明： 应用 readfile()函数时，不需要打开/关闭文件和输出语句，直接应用即可。

☑　file()函数

file()函数将整个文件的内容读入一个数组中。如成功，返回数组，数组中的每个元素都是文件中对应的一行，包括换行符在内；如失败，返回 FALSE。语法如下：

```
array file ( string filename [, int use_include_path [, resource context]] )
```

其参数与 readfile()函数相同，唯一区别是该函数的返回值是数组。

☑　file_get_contents()函数

file_get_contents()函数将文件内容读入一个字符串。如果有 offset 和 maxlen 参数，将在参数 offset 所指定的

位置开始读取长度为 maxlen 的内容。如果失败，返回 FALSE。语法如下：

```
string file_get_contents ( string filename [, bool use_include_path [, resource context [, int offset [, int maxlen]]]] )
```

参数 filename 指定读取的文件名称；参数 use_include_path 控制是否支持在 include_path 中搜索文件，如果支持，则将该值设置为 TRUE。

设计过程

（1）创建 index.php 文件。

（2）在 index.php 文件中分别应用 file()和 file_get_contents()函数读取文本文件中的内容，其关键代码如下：

```
<!--  使用 file()函数读取 count.txt 文件的内容  -->
<?php
$arr = file('test1.txt');
foreach ($arr as $value) {
    echo iconv("gb2312", "utf-8", $value) . "<br>";
}
?>
<!--  使用 file_get_contents()函数读取 count.txt 文件的内容  -->
<?php
$str = file_get_contents('test2.txt');
echo iconv("gb2312", "utf-8", $str);
?>
```

秘笈心法

心法领悟 230：读取整个文件的特性。

在通过 readfile()、file()和 file_get_contents()函数读取整个文件中的内容时，不需要通过 fopen()函数打开文件，也不需要使用 fclose()函数关闭文件。

但是，在读取一个字符、一行字符和任意长度的字符串时，必须应用 fopen()函数打开文件后才能进行读取，在读取完成后还要应用 fclose()函数关闭文件。

实例 231	文本文件的分页读取 光盘位置：光盘\MR\04\231	高级 趣味指数：★★★★☆

实例说明

在遍历文件中的内容时，由于文件内容很多，最理想的方法就是分页读取文本文件中的内容。本实例将介绍如何分页读取文本文件中的数据，其运行结果如图 4.9 所示。

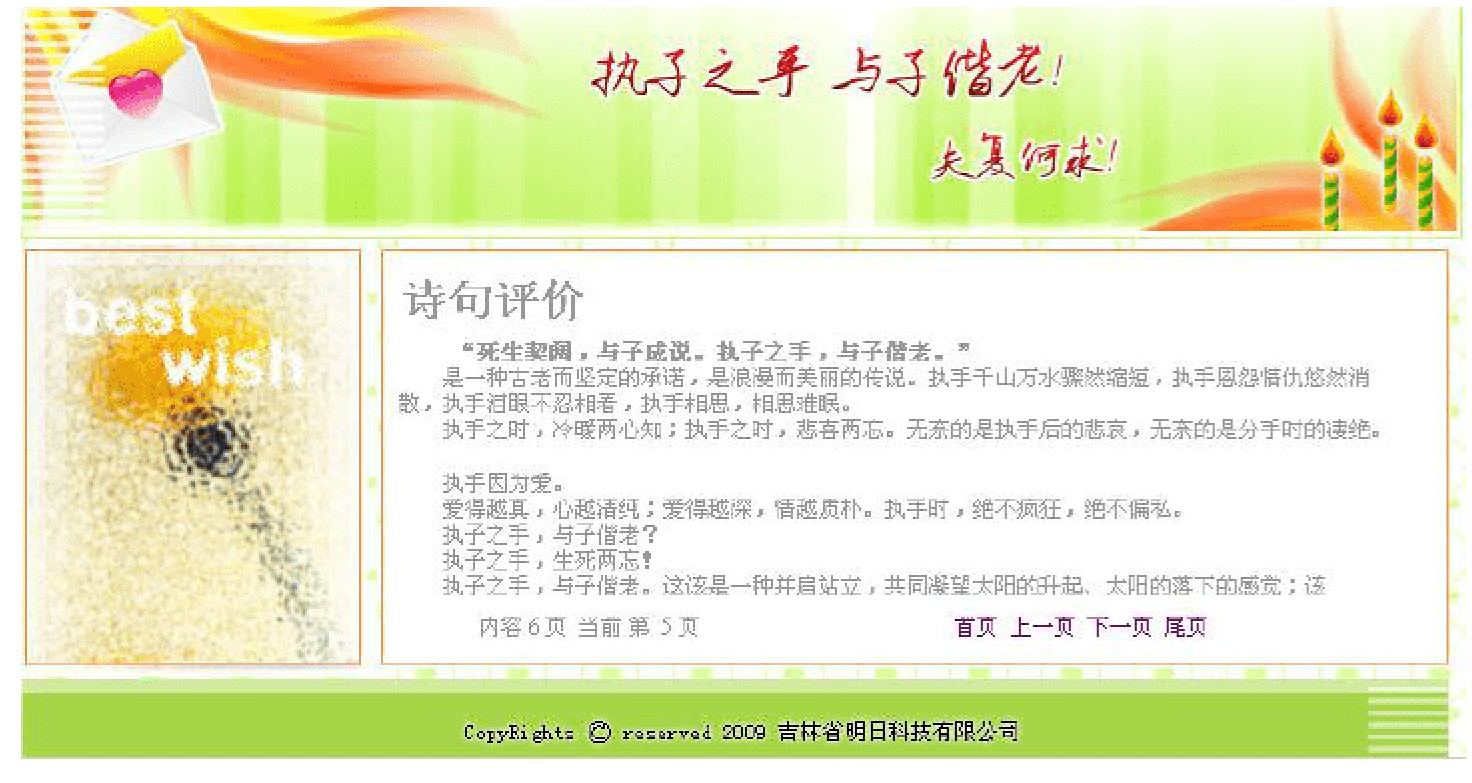
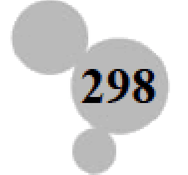


图 4.9　文本文件的分页读取



关键技术

完成超长文本的分页输出需要 3 方面的技术：第一个方面，自定义函数。通过自定义函数读取文本文件，可以避免中文字符串出现乱码；第二个方面，字符串函数。需要通过 strlen()函数计算字符串的长度，通过 substr()函数对字符串进行截取；第三个方面，文件系统函数。通过 file_get_contents()函数读取文本文件中的数据。

自定义函数 msubstr()的语法如下：

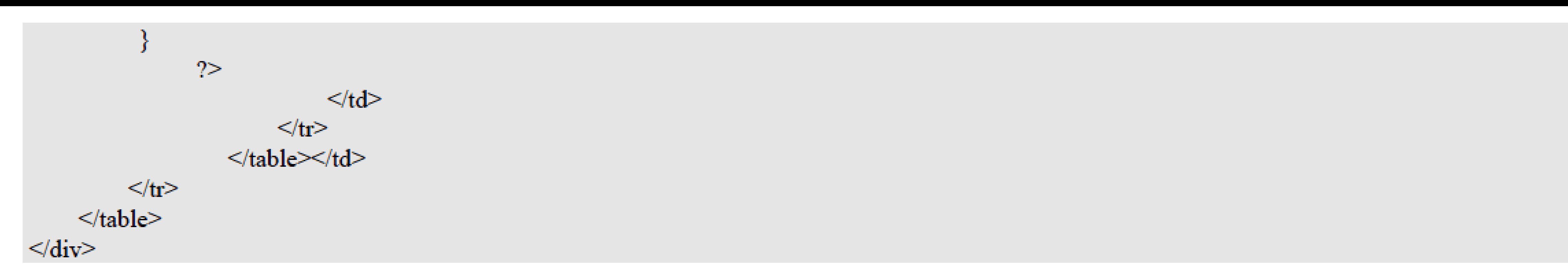
```
//定义一个用于截取一段字符串的函数 msubstr()
function msubstr($str,$start,$len){                //$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
$strlen=$start+$len;                               //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
for($i=0;$i<$strlen;$i++){                         //通过 for 循环语句，循环读取字符串
        if(ord(substr($str,$i,1))>0xa0){           //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
                $tmpstr.=substr($str,$i,2);        //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
                $i++;                              //变量自加 1
        }else{                                     //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
                $tmpstr.=substr($str,$i,1);
}
}
return $tmpstr;                                    //输出字符串
}
```

设计过程

（1）创建 function.php 文件，编写自定义函数 msubstr()，完成对文本文件的截取操作。

（2）创建 index.php 文件，首先通过文件系统函数 file_get_contents()读取整个文件的内容，然后调用自定义函数和字符串函数完成对文件的截取操作，实现截取后内容的分页输出，其关键代码如下：

```
<div id="synopsis">
    <!--创建 div 标签，用于获取 js 文件中返回的分页结果-->
    <table width="545" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td><?php
                        include("function.php");
  //读取超长文本中的数据，实现超长文本中数据的分页显示
  if($_GET['page']){
      $counter=file_get_contents("data.txt");
      $length=strlen($counter);
      $page_count=ceil($length/850);
      $c=msubstr($counter,0,($_GET['page']-1)*850);
      $c1=msubstr($counter,0,$_GET['page']*850);
       echo substr($c1,strlen($c),strlen($c1)-strlen($c));
  }
?>
            </td>
        </tr>
        <tr>
            <td><table width="535" border="0" align="center" cellpadding="0" cellspacing="0">
                    <tr>
                        <!--设置超长文本分页显示的超链接-->
                        <td width="37%" align="center" valign="middle" bgcolor="#FFFFFF">内容 <?php echo $page_count;?> 页
  当前 第 <?php echo $_GET['page'];?> 页</td>
                        <td width="63%" height="28" align="center" valign="middle" bgcolor="#FFFFFF"><?php
              if($_GET['page']!=1){
?>
                            <!--调用 no_refurbish_pagination()函数，实现无刷新的分页输出-->
                            <a href="#" onClick='return no_refurbish_pagination("index_ok.php?page=1")'>首页</a>  <a href="#"
onClick='return no_refurbish_pagination("index_ok.php?page=<?php echo $_GET['page']-1;?>")'>上一页</a>
                            <?php   }
                    if($_GET['page']<$page_count){
?>
                            <a href="#" onClick="return no_refurbish_pagination('index_ok.php?page=<?php echo $_GET['page']+1;?>')">下一
页</a> <a href="#" onClick='return no_refurbish_pagination("index_ok.php?page=<?php echo $page_count;?>")'>尾页</a>
                            <?php
```



```
            }
                ?>
                        </td>
                    </tr>
                </table></td>
            </tr>
        </table>
</div>
```

（3）创建 index_ok.php 文件，实现数据的无刷新分页。index_ok.php 中的内容与 index.php 中<div>标签 synopsis 的内容相同。

（4）创建 js 脚本文件 discuss_js.js，通过 AJAX 实现无刷新的操作。

秘笈心法

心法领悟 231：AJAX 无刷新分页。

在本实例中不但实现了文本文件内容的分页输出，而且是无刷新分页输出。具体操作通过 JS 脚本文件 discuss_js.js 和 index_ok.php 完成。

4.4 文 件 操 作

实例 232	文件操作汇总 光盘位置：光盘\MR\04\232	中级 趣味指数：★★★☆

实例说明

为了便于对网站进行管理、维护和更新，应该设计一个能够对文件进行操作的模块，实现对文件的创建、复制、移动和删除等操作，这样能够给网站的管理工作提供很大方便，不再因为要修改某个文件而登录到 FTP 中、通过下载或上传实现文件的更新，从而节省很多时间。运行本实例，实现对指定文件的创建、复制、移动和删除操作。只要在文本框中输入要复制文件的路径和名称，在对应的文本框中输入指定文件要复制到的具体文件夹的路径和名称（包括指定文件的名称），然后单击“提交”按钮即可，运行结果如图 4.10 所示。

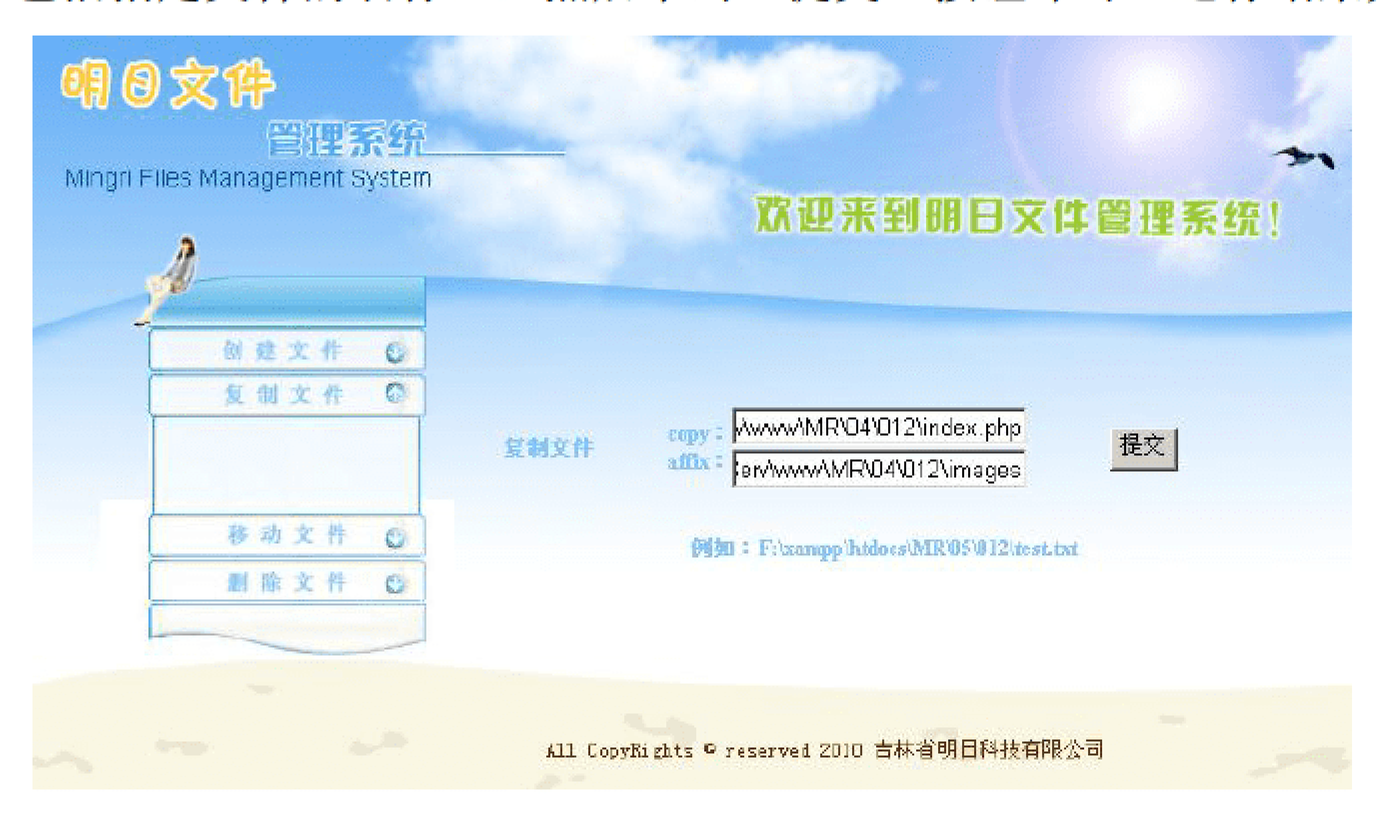
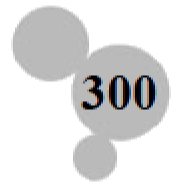


图 4.10 文件操作汇总

关键技术

本实例主要应用 fopen()、copy()、rename()和 unlink()这 4 个函数完成文件的创建、复制、移动和删除操作。



fopen()函数用于打开文件或者 URL，其语法如下：

```
int fopen(string filename, string mode);
```

创建文件主要应用 fopen()函数的 mode 参数，mode 参数的可选值如下：

☑ 'w' 写入方式打开，将文件指针指向文件头并将文件大小截为 0。如果文件不存在，则进行创建。
☑ 'w+' 读写方式打开，将文件指针指向文件头并将文件大小截为 0。如果文件不存在，则进行创建。
☑ 'a' 写入方式打开，将文件指针指向文件末尾。如果文件不存在，则进行创建。
☑ 'a+' 读写方式打开，将文件指针指向文件末尾。如果文件不存在，则进行创建。

说明：上述 4 种方式都可以完成文件的创建操作，前提是指定路径下不存在被创建的文件。

copy()函数主要用于复制文件，其语法如下：

```
int copy(string source, string dest);
```

本函数将文件从 source 复制到 dest，如果成功则返回 TRUE，否则返回 FALSE。

rename()函数主要用于文件重命名，其语法如下：

```
bool rename(string oldname,newname);
```

本函数将 oldname 重命名为 newname，如果成功则返回 TRUE，否则返回 FALSE。

说明：ename()函数不但可以给文件重命名，当将指定的文件移动到另外一个路径下的文件夹中时，不改变该文件的名称，即可实现对指定文件的移动操作。

unlink()函数主要用于删除文件，其语法如下：

```
bool unlink(string filename);
```

该函数用于删除文件，如果删除成功则返回 TRUE，否则返回 FALSE。

设计过程

（1）创建 index.php 文件。首先，设计文件汇总的页面。然后应用 switch 语句设计一个简单的框架，完成复制、移动、创建和删除操作之间的切换。最后，编写 PHP 脚本，根据表单中提交的数据，执行不同的文件操作，其关键代码如下：

```
<?php
switch ($_GET ['operate']) {
	case "found" :
		include ("found.php");
		break;
	case "copy" :
		include ("copy.php");
		break;
	case "move" :
		include ("move.php");
		break;
	case "delete" :
		include ("delete.php");
		break;
	default :
		include ("copy.php");
}
?>
<?php
if ($_POST ['Submit'] == "复制") {
$copy = iconv ( "utf-8", "gb2312", $_POST ['copys'] );
$copys2 = iconv ( "utf-8", "gb2312", $_POST ['copys2'] );
if (copy ( $copy, $copys2 )) {
	echo "<script>alert('复制成功!!');</script>";
} else {
	echo "<script>alert('复制失败!!');</script>";
}
}
```

```
//省略了部分代码
?>
```

（2）分别创建 copy.php、delete.php、found.php 和 move.php 文件，完成不同操作的执行页面。copy.php 文件的代码如下：

```
<table width="466" height="112" border="0" cellpadding="0" cellspacing="0">
<form name="form1" method="post" action="">
<tr>
        <td width="111" rowspan="2" align="center">复制文件</td>
        <td width="67" align="right" valign="bottom">copy：</td>
        <td width="229" height="39" valign="bottom"><input name="copys"
                type="text" id="copys" size="22"></td>
        <td width="126" rowspan="2" align="left" valign="middle"><input
                type="submit" name="Submit" value="复制"></td>
</tr>
<tr>
        <td align="right" valign="top">affix：</td>
        <td height="41" valign="top"><input name="copys2" type="text"
                id="copys2" size="22"></td>
</tr>
<tr>
        <td colspan="4" align="center">例如：F:\xampp\htdocs\MR\05\012\test.txt</td>
</tr>
</form>
</table>
```

秘笈心法

心法领悟 232：编码格式转换的必要性。

在应用文件操作函数执行文件的创建、移动、复制和删除操作时，由于实例的页面编码格式是 UTF-8，所以从表单中获取的元素值不能直接应用到文件操作函数中，必须对其进行编码转换，由 UTF-8 编码转换为 GB2312 编码，否则文件操作函数不能正确地执行。

实例 233 目录、文件定位器

光盘位置：光盘\MR\04\233

中级

趣味指数：★★★★☆

实例说明

对目录、文件的操作不应该是盲目的，应该首先确定其是否存在，如果存在才能对它执行各种操作，否则是没有任何意义的。相反，有些操作必须是在指定的目录、文件不存在的情况下进行的。此时必须有一个方法能够对目录、文件进行定位，确定它是否存在，这就是本实例将要讲解的内容。运行本实例，在文本框中输入目录、文件的完整路径，单击“提交”按钮，运行结果如图 4.11 所示。



图 4.11　目录、文件定位器



说明：本实例实现的功能虽然非常简单，但是这个方法在很多程序中都会被用到，只是表现的方式不同。

关键技术

PHP 中判断指定的目录、文件是否存在，应用的是 file_exists()函数，该函数的语法如下：

```
int file_exists(string filename);
```

该函数用来检测目录、文件是否存在，如果存在返回 TRUE，否则返回 FALSE。

说明：要在 Windows 下检查网络中的共享文件应该使用//computername/share/filename 或者\\computername\share\filename。

设计过程

（1）创建 index.php 文件，设计页面布局。

（2）添加表单，设置文本框，提交目录、文件的完整路径，并设置提交按钮。

（3）编写 PHP 脚本，获取表单中提交的目录、文件路径。首先对获取的元素值进行编码格式的转换，然后应用 file_exists()函数判断指定的目录、文件是否存在，其关键代码如下：

```
<?php
if ($_POST ['file_name'] != "") {
	$file_name = iconv ( "utf-8", "gb2312", $_POST ['file_name'] );
	if (file_exists ( $file_name )) {
		echo "<script>alert('目录、文件已定位！');</script>";
	} else {
		echo "<script>alert('目录、文件不存在！');</script>";
	}
} else {
	echo "<script>alert('请输入正确的目录、文件路径！');</script>";
}
?>
```

秘笈心法

心法领悟 233：UTF-8 编码格式对中文字符串的影响。

在本实例中，由于使用的是 UTF-8 编码格式，因此当提交的目录、文件路径中存在中文字符串时，file_exists()函数就不能正确地判断目录、文件是否存在。只有将 UTF-8 编码格式的元素值转换为 GB2312 编码格式之后，file_exists()函数才能对中文字符串的目录、文件路径作出正确的判断。

实例 234	改头换面 光盘位置：光盘\MR\04\234	初级 趣味指数：★★☆

实例说明

在对网站进行管理和维护的过程中，经常会修改文件的名称。本实例将介绍一种修改文件名称的方法。运行本实例，在“原文件名称”文本框中输入原始文件的存储路径和文件名，在“新文件名称”文本框中输入新文件的路径和文件名，单击“提交”按钮，运行结果如图 4.12 所示。



图 4.12　修改文件名

关键技术

文件重命名应用的是 rename()函数，不过首先要应用 file_exists()函数判断该文件是否存在，如果存在则应用 rename()函数对该文件进行重命名，否则给出提示信息。

设计过程

（1）创建 index.php 文件，设计页面布局。

（2）添加表单，设置两个文本框，提交文件的原始名称和新名称，并设置提交按钮。

（3）编写 PHP 脚本，获取表单中提交的元素值。由于网页使用的是 UTF-8 编码，所以为了避免在获取中文字符串时出现乱码，应先用 iconv()函数对获取的元素值进行编码格式的转换，然后应用 file_exists()函数判断指定的文件是否存在，如果存在则应用 rename()函数对其进行重新命名，其关键代码如下：

```
<?php
if ($_POST['Submit'] == "提交") {
        $old_name=iconv("utf-8","gb2312",$_POST['old_name']);                 //获取表单元素，并进行编码格式转换
        $new_name=iconv("utf-8","gb2312",$_POST['new_name']);                 //获取表单元素，并进行编码格式转换
        if (file_exists ( $old_name )) {                                      //判断文件是否存在
                if (rename ( $old_name, $new_name )) {                        //把原文件重新命名
                        echo "修改成功!";
                }
        } else {
                print $_POST['old_name'] . "文件不存在!<br>";
        }
}
?>
```

秘笈心法

心法领悟 234：iconv()函数在转码过程中的重要性。

如果程序设计时应用的是 UTF-8 编码，那么就有必要了解 iconv()函数。因为操作系统默认使用的是简体中文（GB2312 编码或者 GBK 编码），所以在将 UTF-8 编码格式的数据应用到默认为 GB2312 编码的程序中时，难免会执行失败，特别是涉及中文字符串时。例如，在本实例中，如果直接将获取的表单元素值应用到 file_exists()函数或者 rename()函数中，就会出现乱码。

实例 235	文件属性分析 光盘位置：光盘\MR\04\235	中级 趣味指数：★★★★

实例说明

如果说判断目录、文件是否存在是对目录、文件进行操作的前提条件，那么获取文件属性则是对文件进行



操作的必要条件。因为在执行一些特殊的操作之前，必须对文件的类型、大小或者修改时间进行判断。在本实例中，将介绍如何获取文件的类型、大小和修改时间。运行本实例，在文本框中输入正确的目录或者文件的路径，单击“提交”按钮，运行结果如图 4.13 所示。



图 4.13　文件属性分析

关键技术

在 PHP 中，获取文件类型使用 filetype()函数，获取文件大小使用 filesize()函数，而获取修改时间使用 filemtime()函数。

filetype()函数用于获取文件类型，其语法如下：

```
string filetype ( string filename )
```

该函数返回文件的类型，类型值包括 fifo、char、dir、block、link、file 和 unknown。如果出错则返回 FALSE。

filesize()函数用于获取文件大小，其语法如下：

```
int filesize ( string filename )
```

该函数返回文件大小的字节数，如果出错返回 FALSE （在错误报告级别为 E_WARNING 的情况下）。

说明： 因为 PHP 的整数类型是有符号的，并且大多数平台使用 32 位整数，filesize() 函数在碰到大于 2GB 的文件时可能会返回非预期的结果。对于 2GB~4GB 之间的文件通常可以使用 sprintf("%u", filesize($file)) 来克服此问题。

filemtime()函数用于获取文件修改时间，其语法如下：

```
int filemtime ( string filename )
```

该函数返回文件上次被修改的时间，出错时返回 FALSE。时间以 UNIX 时间戳的方式返回，可用于 date()函数。

设计过程

（1）创建 index.php 文件，设计页面布局。

（2）添加表单，设置目录、文件提交的文本框，提交目录或者文件的路径，并设置提交按钮。

（3）编写 PHP 脚本，获取表单中提交的元素值。首先，判断提交的值是否为空，如果不为空，则应用 iconv()函数对字符串的编码格式进行转换。然后，应用 file_exists()函数判断指定的目录或者文件是否存在，如果存在则获取目录或者文件的类型、大小以及修改时间，其关键代码如下：

```
<?php
    if ($_POST ['file_name'] != "") {                                    //判断表单提交的值是否为空
        $file_name = iconv ( "utf-8", "gb2312", $_POST ['file_name'] );  //完成编码格式的转换
        if (file_exists ( $file_name )) {                                //判断目录或者文件是否存在
            $file_type = filetype ( $file_name );                        //获取文件类型
            echo "<br>文件类型：" . $file_type . "<br>";
            if ($file_type != "dir") {                                   //判断如果不是目录
                $file_size = filesize ( $file_name );                    //获取文件的大小
                echo "文件大小：" . $file_size . " 字节" . "<br>";
            }
```

能。本实例将演示如何通过 phpMyAdmin 修改 MySQL 用户密码。

关键技术

通过 phpMyAdmin 修改 MySQL 用户密码的方法有两种：

（1）直接通过 phpMyAdmin 的界面进行更改。

（2）通过修改 phpMyAdmin 文件夹下的配置文件 config.php 实现更改。

设计过程

1. 通过 phpMyAdmin 的界面进行密码修改

（1）登录到 phpMyAdmin 图形化工具页面，单击“修改密码”按钮，如图 8.119 所示。

（2）根据页面提示输入新密码和确认密码，如图 8.120 所示。

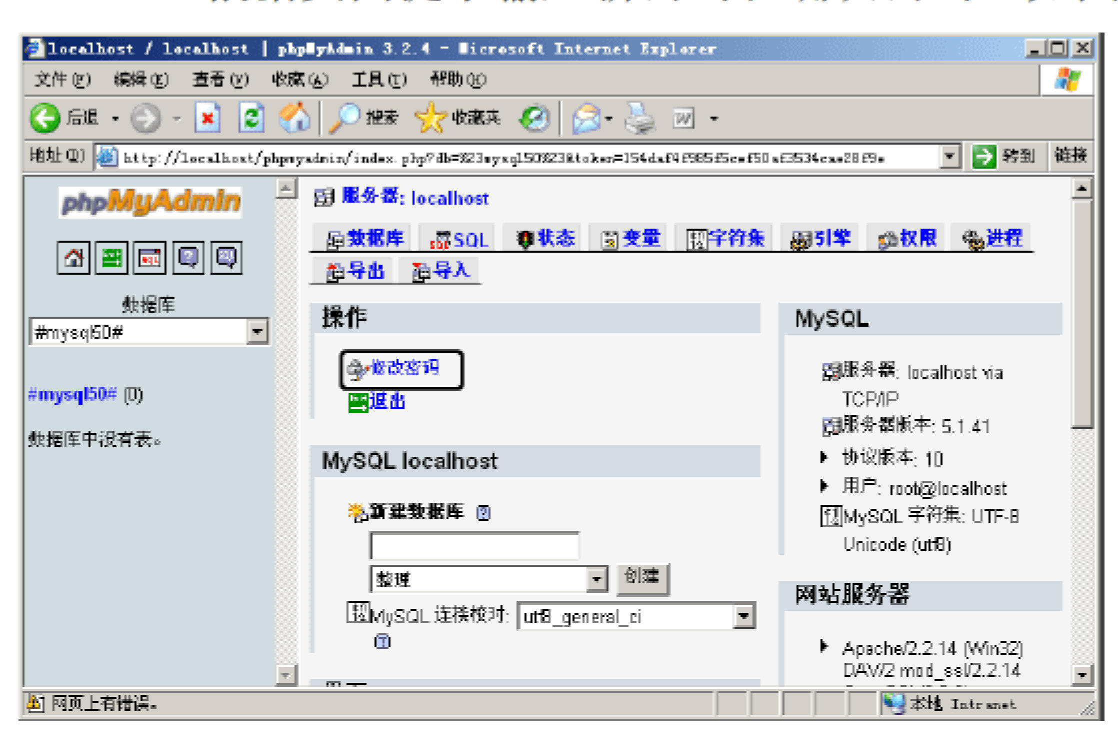


图 8.119　phpMyAdmin 图形化界面

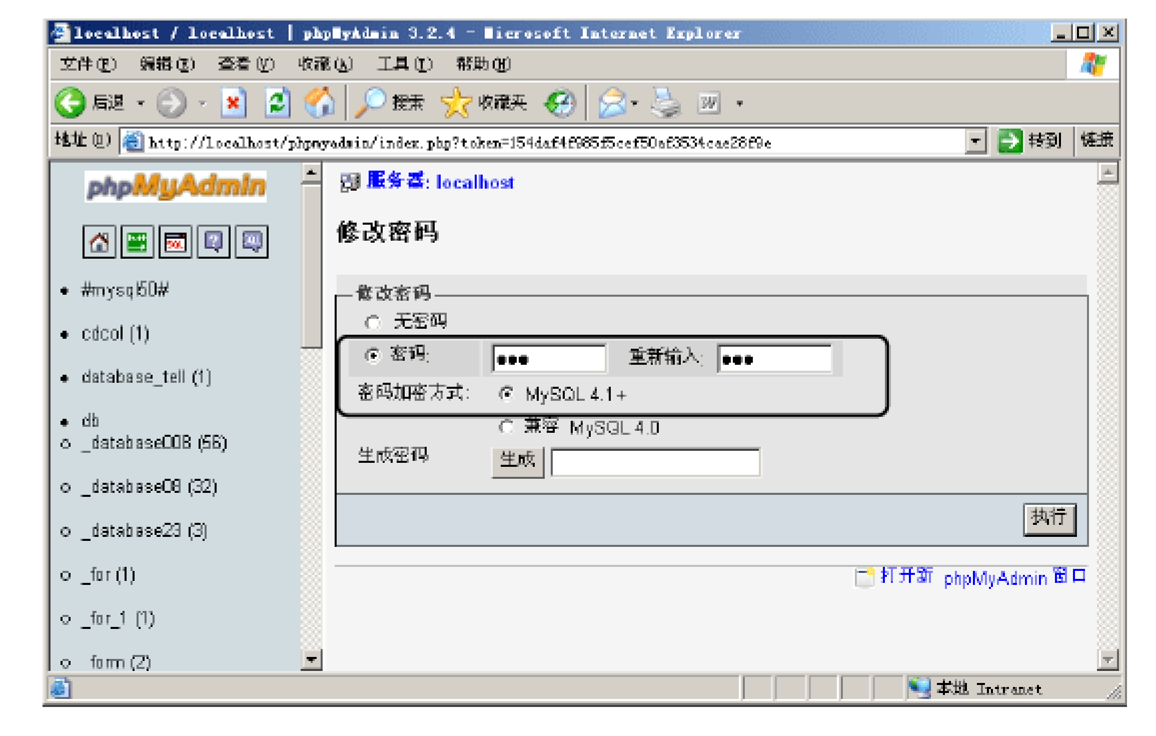


图 8.120　修改 MySQL 数据库密码

（3）单击“执行”按钮，修改 MySQL 数据库密码，如图 8.121 所示。

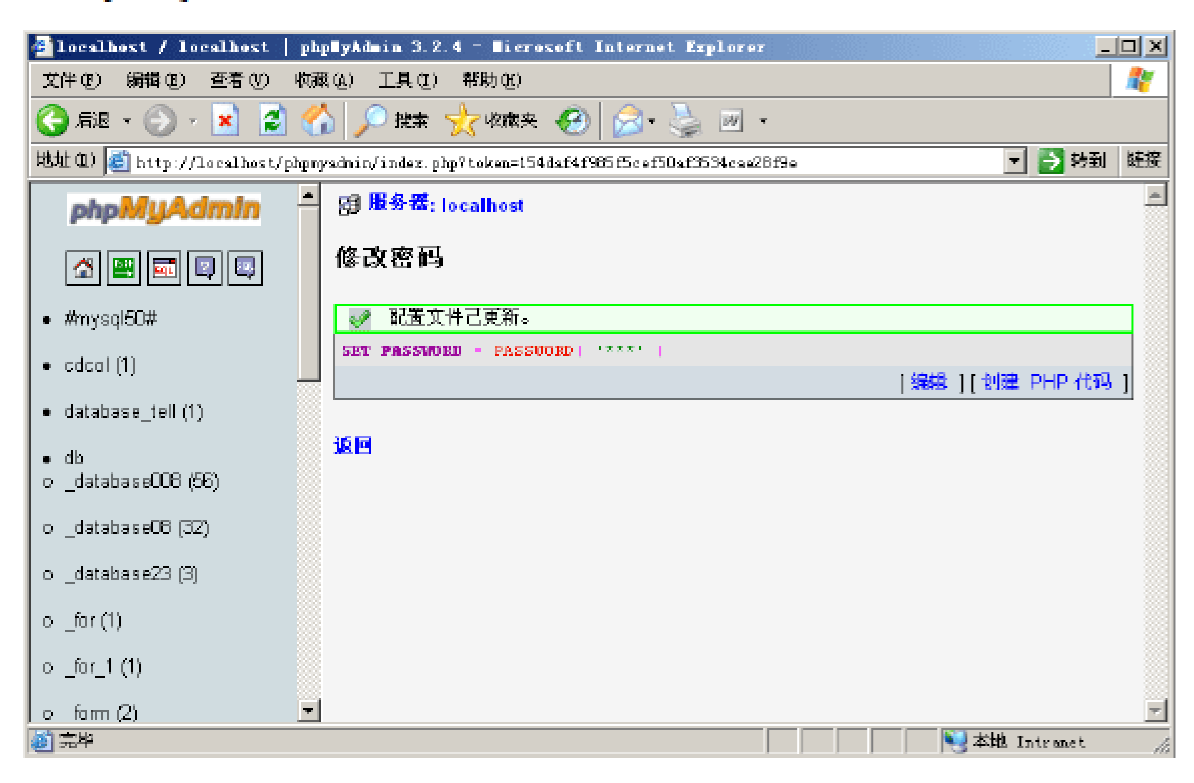


图 8.121　完成密码修改

2. 通过修改配置文件进行密码修改

（1）将下载的 phpMyAdmin-x.zip 文件解压到 Apache 默认的根目录下，解压后的名称是 phpMyAdmin-2.x.x.x，其中的 2.x.x.x 是版本号，为方便使用，可以将其重新命名为 phpMyAdmin。

（2）打开 phpMyAdmin 文件夹，找到配置文件 config.php，然后使用记事本或其他编辑工具打开该文件，找到该文件中的以下内容进行配置：

```
$cfg['PmaAbsoluteUri'] = 'http://127.0.0.1/phpMyAdmin';
```

以上用于设置 phpMyAdmin 的 URL，例如 http://127.0.0.1/phpMyAdmin。

```
$cfg['blowfish_secret'] = 'root';
```



秘笈心法

心法领悟 446：设置统一的编码格式。

在创建数据表字段时，一般将字符类型的数据字段设置为统一的编码格式，对于数值型数据字段，可以不进行编码格式的设置。

实例 447	phpMyAdmin 操作数据库 光盘位置：光盘\MR\08\447	高级 趣味指数：★★★★

实例说明

phpMyAdmin 可以实现在 cmd 命令行下对数据库的操作，而且操作更简单、直观。本实例通过 phpMyAdmin 实现对数据库的创建和删除操作。

关键技术

切记，在创建数据库时，一定要设置数据库的编码格式。注意，一旦执行删除操作，数据库将不能再恢复。

设计过程

（1）登录到 phpMyAdmin 主页，根据提示编写新建数据库的名称并设置数据库的编码格式，如图 8.124 所示。

（2）单击“创建”按钮，创建数据库并自动选择创建的数据库为当前数据库。

（3）如果数据库已经没有任何作用，可以通过 phpMyAdmin 删除数据库。选择想要删除的数据库，单击导航栏中的“删除”按钮，根据提示决定是否删除数据库，如图 8.125 所示。



图 8.124　创建数据库

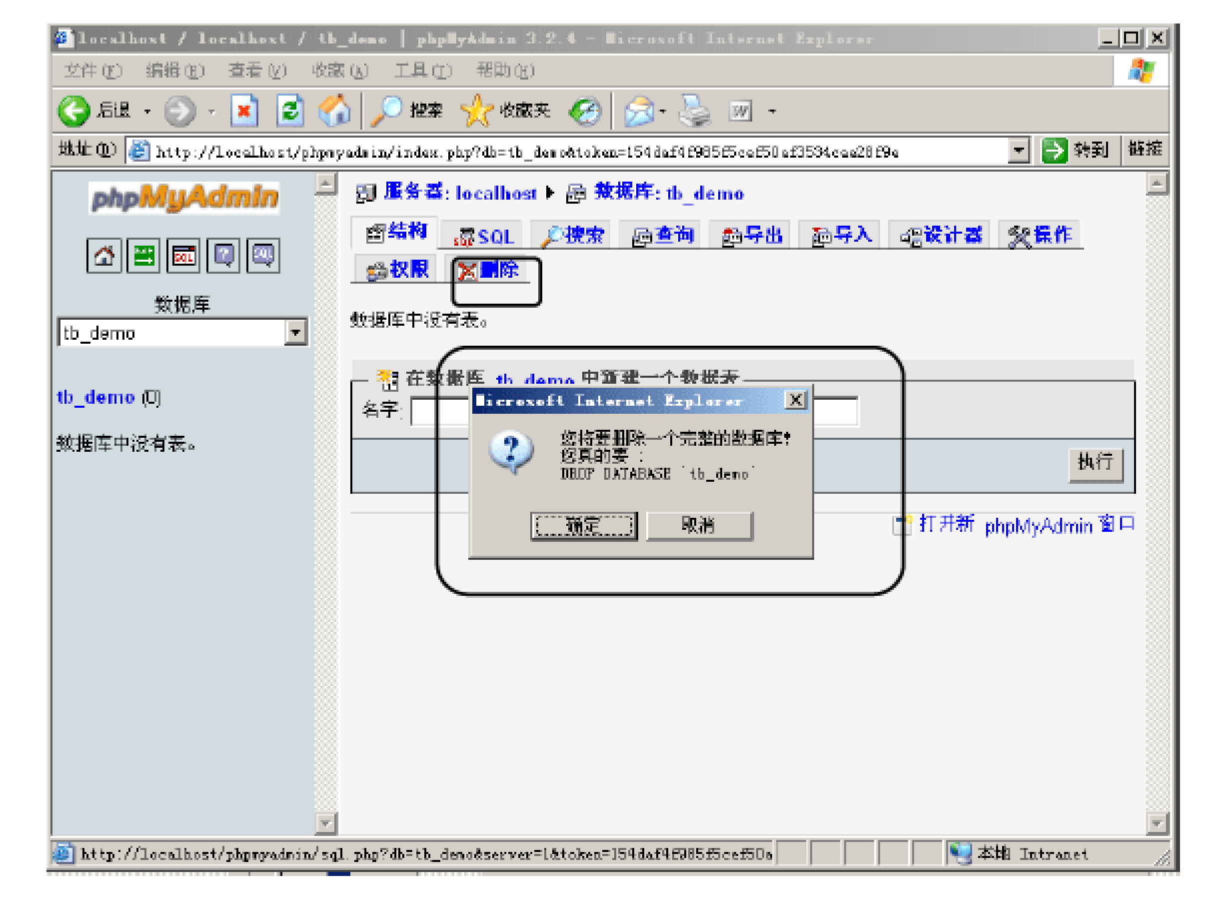
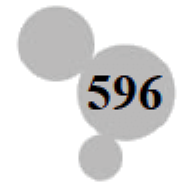


图 8.125　删除数据库

秘笈心法

心法领悟 447：phpMyAdmin 图形化工具与 cmd 命令下操作数据库。

通过 phpMyAdmin 图形化工具与在 cmd 命令下操作数据库没有任何区别，只是 phpMyAdmin 的设计者对 MySQL 数据库的相关指令进行了封装，使一些对 MySQL 数据库不了解的用户也可以操作数据库。



实例 448

phpMyAdmin 操作数据表

光盘位置：光盘\MR\08\448

高级

趣味指数：★★★★

实例说明

对数据表的操作一般包括创建和删除，本实例介绍应用 phpMyAdmin 图形化工具实现对数据表的创建和删除操作。

关键技术

想要创建数据表，首先要创建数据库，当数据库创建完成时自动选择已经创建好的数据库，根据提示编写数据表名称并创建数据表。而删除数据表时，需要选择想要删除的表，然后单击“删除”按钮即可完成。

设计过程

（1）登录到 phpMyAdmin 界面，创建数据库，选择数据库编码格式，单击“创建”按钮，根据提示编写数据表名称，并填写想要创建数据表的字段数目，如图 8.126 所示。

（2）进入编写数据库字段页，建立数据表结构，填写字段信息，其中包括字段名称、字段类型、字段长度、是否允许为空等，如图 8.127 所示。

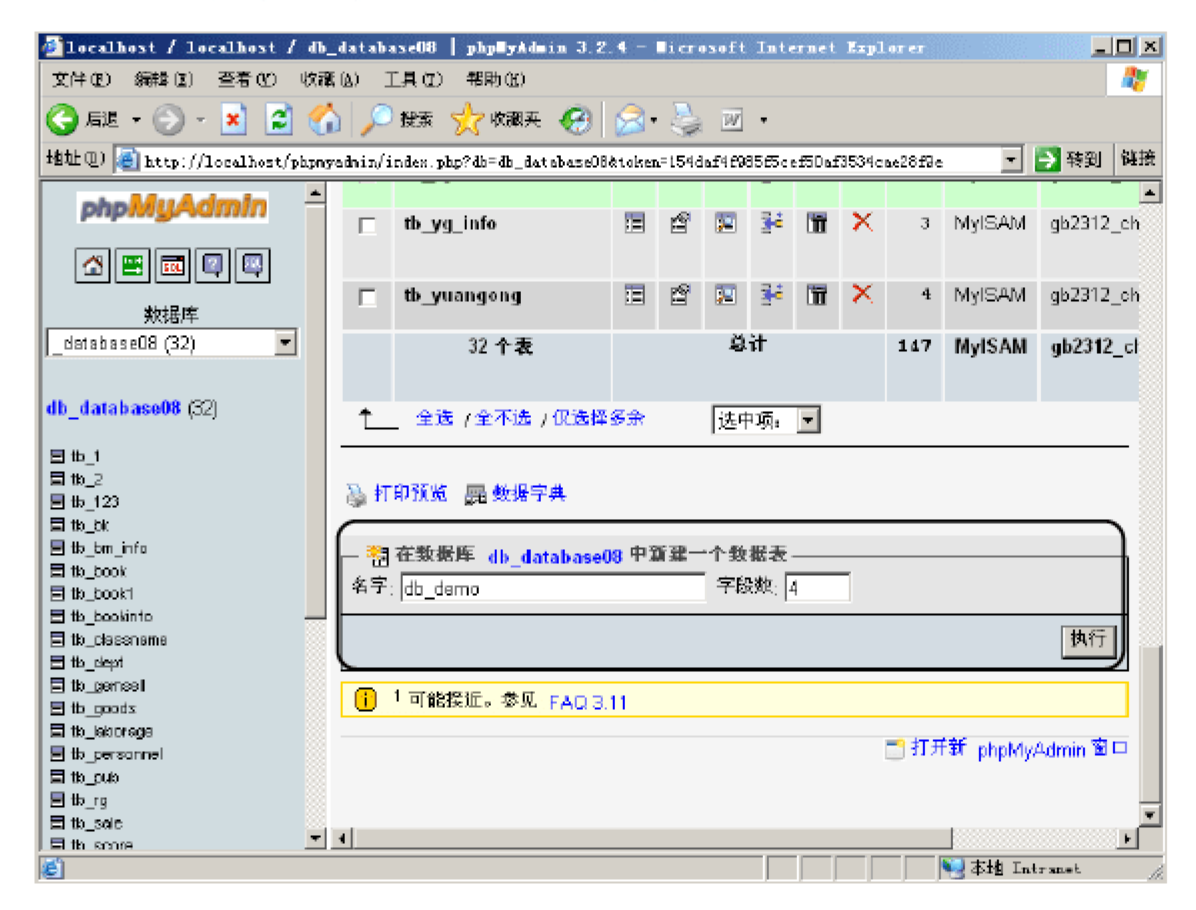


图 8.126　创建数据表

图 8.127　创建数据表结构

（3）单击“保存”按钮，完成数据表的创建。

（4）删除数据表时，首先选择要删除的数据表名称并进入数据表操作界面，单击导航栏中的“删除”按钮，根据提示删除数据表，如图 8.128 所示。

秘笈心法

心法领悟 448：使用 phpMyAdmin 图形化工具创建数据表的注意事项。

在创建数据表结构时也要对数据表字段进行编码格式设置，而且必须为数据表指定主键、设置索引，否则会出现如图 8.129 所示的提示内容。

（2）插入数据信息。单击导航栏中的“插入”按钮，进入插入信息界面，完成数据的添加后单击“执行”按钮，实现插入数据，如图 8.131 所示。

（3）删除数据信息。首先返回到数据浏览页，如果要删除单条数据信息，直接单击该条数据的删除按钮；如果要删除多条数据信息，可以选中数据信息前端的复选框，然后单击数据表信息下端的“选中项删除”按钮；如果要删除数据表中的所有信息，可以单击导航栏中的“清空”按钮，实现删除所有数据信息，如图 8.132 所示。

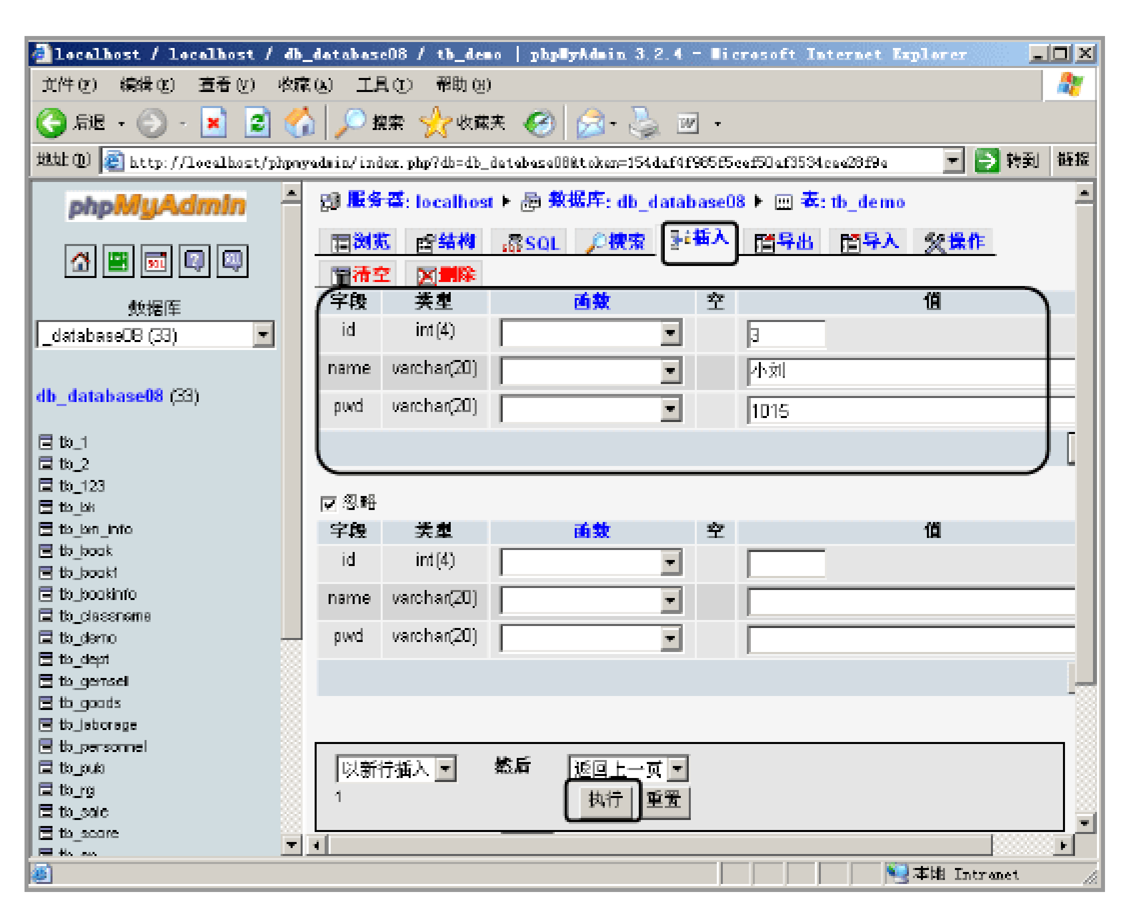


图 8.131　插入数据信息

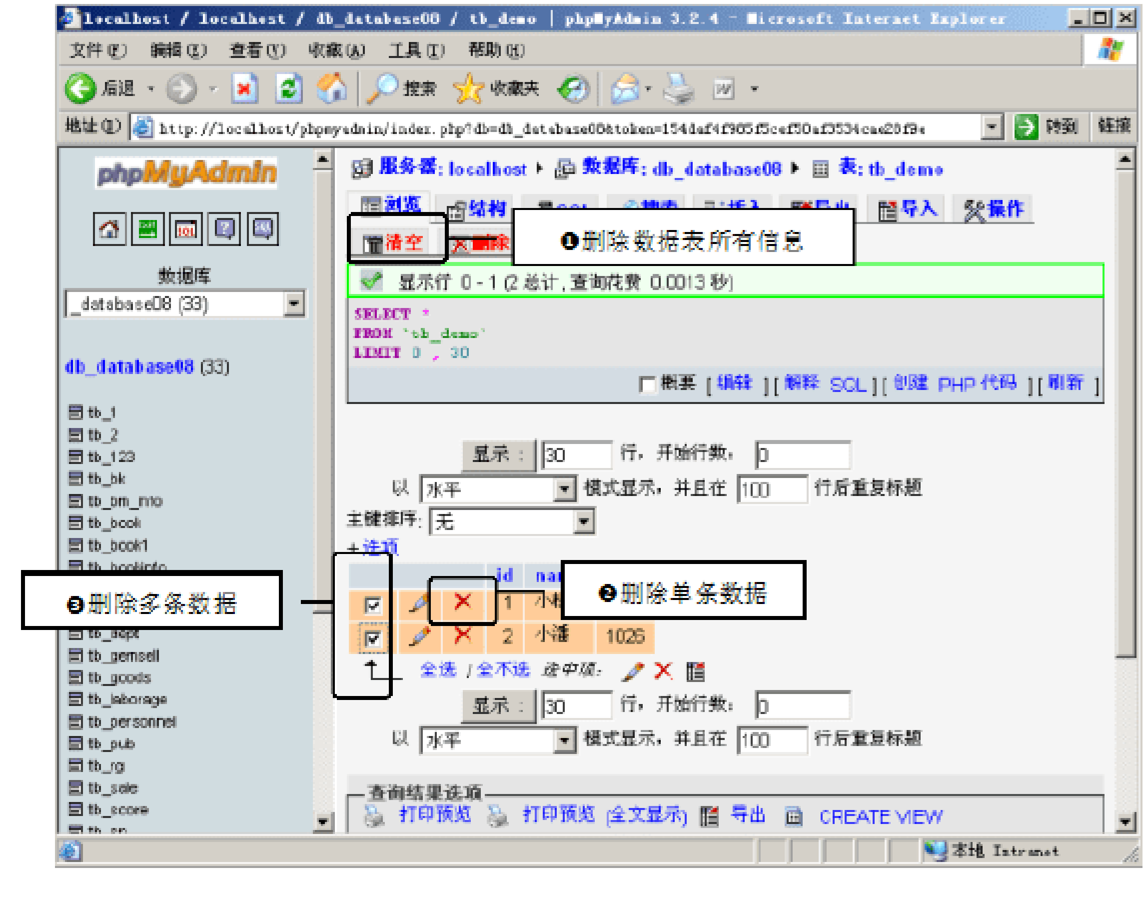


图 8.132　删除数据操作

（4）修改数据信息。修改数据信息与删除数据信息类似，可以分为修改单条数据信息和修改多条数据信息。修改单条数据信息时直接单击该条数据的修改按钮进入修改页面，即可实现修改；修改多条数据信息时，首先要选中复选框，然后单击“选中项修改”按钮，进入修改页面完成修改，如图 8.133 和图 8.134 所示。

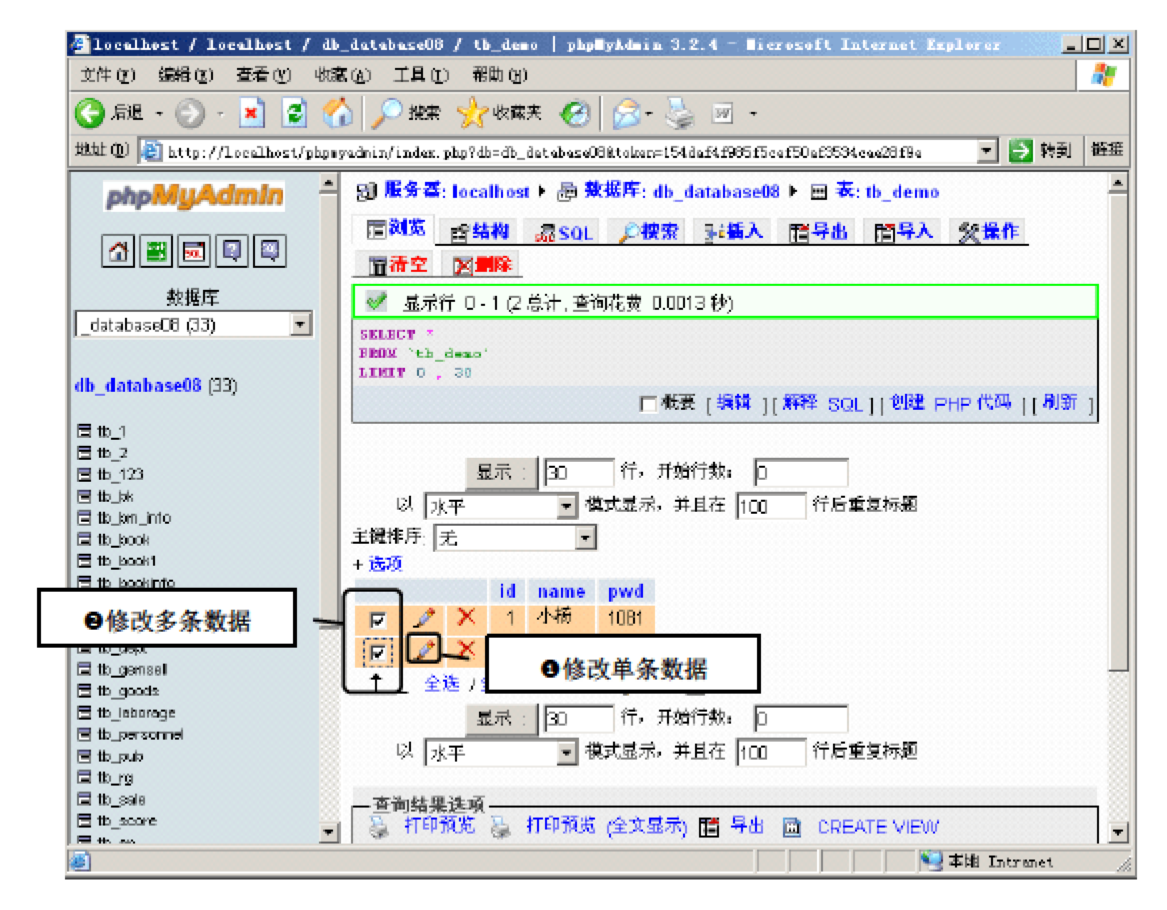


图 8.133　修改数据信息

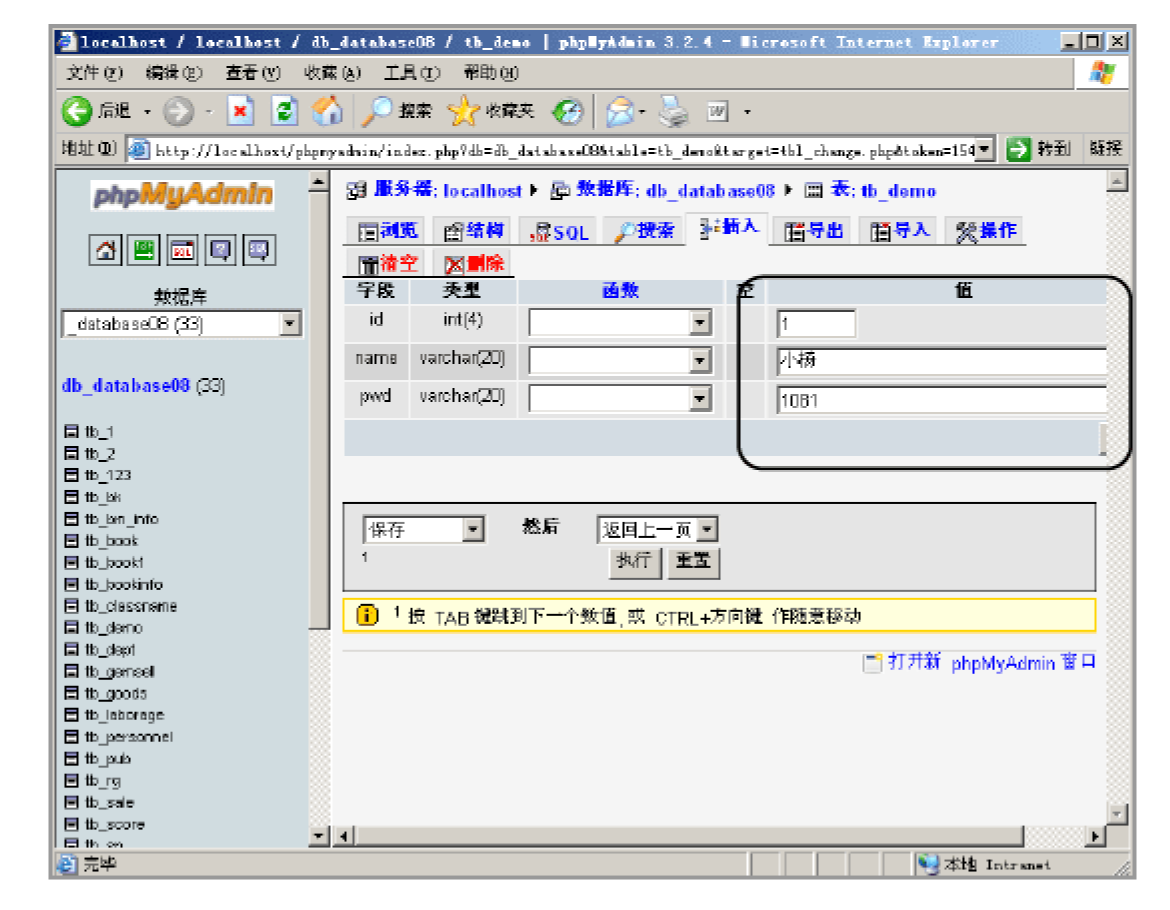


图 8.134　更改数据信息页

秘笈心法

心法领悟 449：在 phpMyAdmin 图形化界面工具中修改操作与删除操作的区别。

如果在修改多条数据时，误选中不需更改的数据信息前的复选框，进入修改页面后可以不对此条数据做任何修改，此条数据信息不会有任何变化。如果在删除多条数据时，误选中了不需删除的数据信息前的复选框，则单击“执行”按钮后信息会被删除。



9.1　PHP 的 MySQLi 扩展库

在 PHP 5.0 以后的版本中，为适应 MySQL 数据库的新特性重新编写了 MySQLi 扩展库。MySQLi 扩展库同时提供了面向过程和面向对象的两种调用接口，本节将通过具体实例讲解如何使用 MySQLi 扩展库对 MySQL 数据库进行操作。

实例 450	使用 MySQLi 扩展库连接 MySQL 数据库 光盘位置：光盘\MR\09\450	中级 趣味指数：★★★

实例说明

本实例主要实现使用 MySQLi 扩展库的面向对象的方式建立与 MySQL 数据库的连接。运行本实例，如图 9.1 所示，在页面中打印出数据库连接成功的提示信息，从而说明已经成功连接上 MySQL 数据库。



图 9.1　数据库连接成功

关键技术

使用 MySQLi 扩展库面向对象的方法建立与 MySQL 数据库的连接，主要是使用 mysqli 类实现的，该类的构造方法的语法格式如下：

```
class mysqli{
    __construct ( [string host [, string username [, string passwd [, string dbname [, int port]]]]] )
}
```

mysqli 类的构造方法的参数说明如表 9.1 所示。

表 9.1　mysqli 类的构造方法的参数说明

参 数 名 称	说　　明
host	MySQL 数据库服务器名称
username	MySQL 数据库服务器用户名
passwd	MySQL 数据库服务器密码
dbname	要连接的 MySQL 数据库的名称
port	MySQL 数据库服务器端口号，默认为 3306

设计过程

（1）创建数据库 db_database09。在 MySQL 中创建数据库可以通过 SQL 语句或 phpMyAdmin 图形化管理工具实现，本实例将采用命令方式创建 MySQL 数据库，其创建代码如下：

```
CREATE DATABASE db_database09 ;
```

（2）使用函数 parse_ini_file()解析配置文件 config.ini 以获得数据库连接参数，然后使用 new 关键字对 mysqli



类进行实例化，最后使用函数 mysqli_connect_errno()判断是否成功连接上了 MySQL 数据库，实现该过程的代码如下：

```
try{
    $config = parse_ini_file(realpath(dirname(__FILE__) . '/config/config.ini'));         //解析 config.ini 文件
    $mysqli = new mysqli($config['host'], $config['username'], $config['password'], $config['dbname']);   //对 mysqli 类进行实例化
    if(mysqli_connect_errno()){                                                            //判断是否成功连接上 MySQL 数据库
        throw new Exception('<br/>数据库连接错误！ ');                                      //如果连接错误，则抛出异常
    }else{
        echo '数据库连接成功！ ';                                                           //打印连接成功的提示
    }
}catch (Exception $e){                                                                     //捕获异常
    echo $e->getMessage();                                                                 //打印异常信息
}
```

注意：使用 MySQLi 扩展库完成对 MySQL 数据库的操作后，还应该通过调用 MySQLi 类实例对象的 close()方法释放资源。

秘笈心法

心法领悟 450：使用 MySQLi 扩展库的面向过程的方式连接 MySQL 数据库。

使用 MySQLi 扩展库不仅可以采用面向对象的方式建立与 MySQL 数据库的连接，为了适应不同开发习惯用户的需求，同时提供了面向过程的方式对 MySQL 数据库进行管理。使用 MySQLi 面向过程的方式建立与 MySQL 数据库的连接是通过 mysqli_connect()函数实现的，该函数的语法格式如下：

```
mysqli_connect( [string host [, string username [, string passwd [, string dbname [, int port]]]]] )
```

该函数参数的使用方式与 mysqli 类的构造方法相同，这里不再赘述。

实例 451 通过 MySQLi 扩展库实现数据库信息的检索

光盘位置：光盘\MR\09\451

初级

趣味指数：★★★

实例说明

在 Web 项目开发过程中，经常需要对大量数据信息进行查询操作，本实例主要介绍如何使用 MySQLi 扩展技术实现商品信息的检索。运行本实例，如图 9.2 所示，在图中文本框中输入要查找的商品名称，单击“查找”按钮，如果该商品存在则显示该商品的所有信息，否则提示没有查找到该商品。

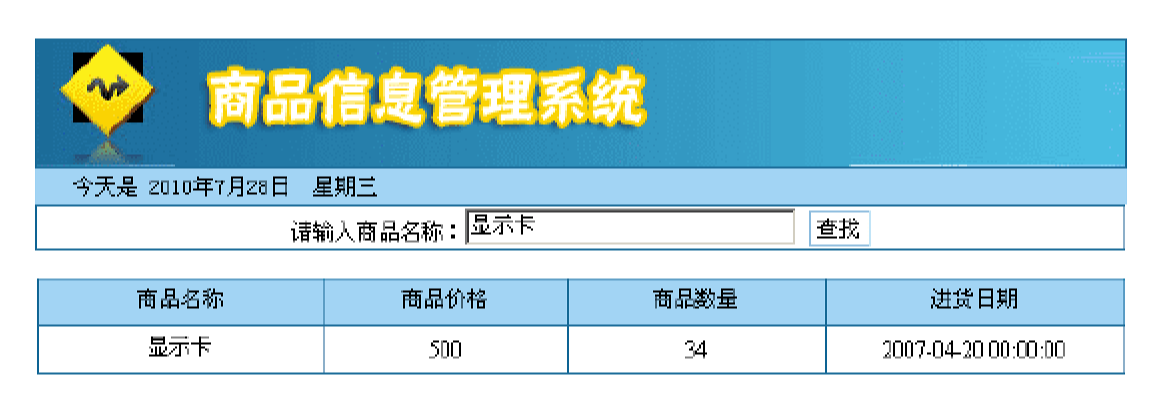


图 9.2 商品信息查询

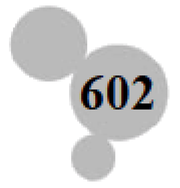
关键技术

在 MySQLi 扩展库面向对象的编程方式中，可以使用 mysqli 类中的 query()方法执行查询操作，该方法的语法格式如下：

```
mixed query ( string query [, int resultmode] )
```

参数说明：

query：要执行的查询语句。



实例 452　查看服务器连接错误报告

光盘位置：光盘\MR\09\452

初级

趣味指数：★★★

实例说明

在开源项目开发中，MySQL 数据库被广泛应用。这不但因为 MySQL 数据库是完全免费的，而且 MySQL 数据库和 PHP 一样都是完全跨平台的。但在实际项目开发过程中，PHP 与 MySQL 数据库产生连接错误是不可避免的，本实例将讲解如何实现尽可能早地发现这种连接错误。运行本实例，分别如图 9.3 和图 9.4 所示。图中所示错误都是当连接数据库服务器时用户密码输入错误所致，其中图 9.3 在页面中打印出上述错误的信息，图 9.4 在页面中打印出上述错误的错误号。

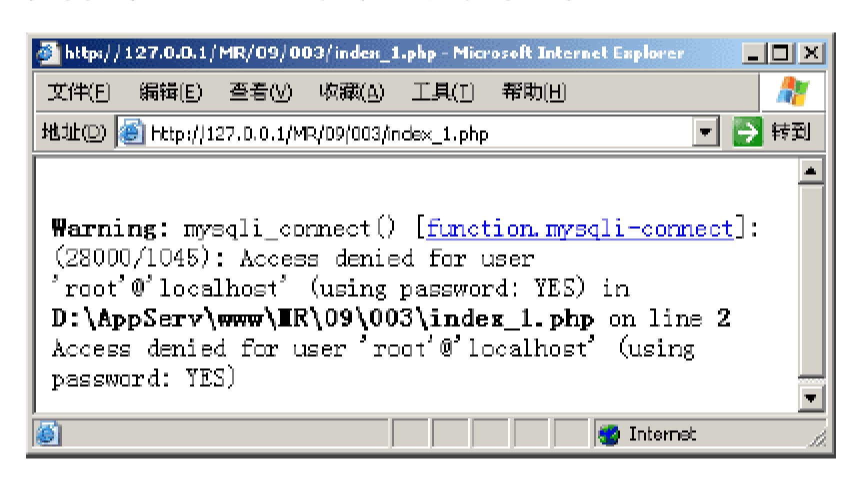


图 9.3　显示错误信息

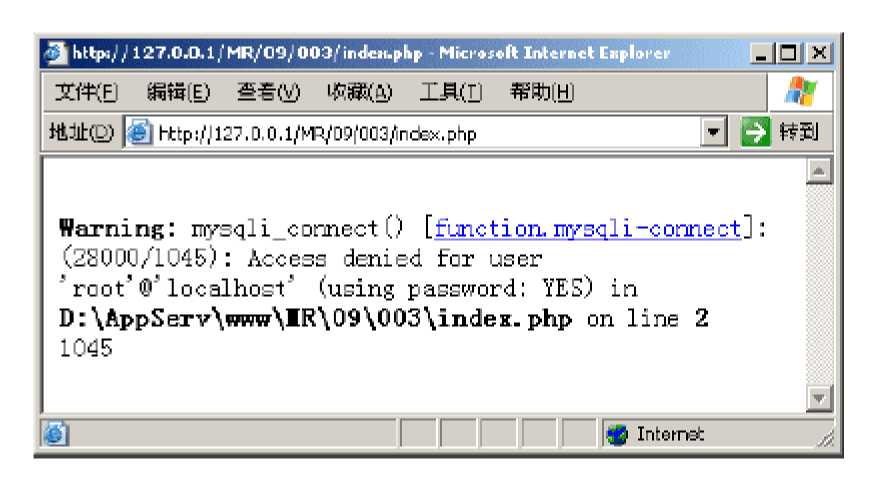


图 9.4　显示错误号

关键技术

在 MySQLi 扩展技术中，可以使用两种方式获得数据库连接错误报告，一种是直接获取错误的名称，另一种是获取连接错误所代表的错误号。

直接获取错误的名称可以使用函数 mysqli_connect_error()实现，该函数的语法格式如下：

```
string mysqli_connect_error(mysqli link)
```

该函数将返回一个错误消息，该消息对应于使用 MySQLi 扩展技术连接 MySQL 数据库时所产生的错误。

获取连接 MySQL 数据库时产生的错误号可以使用函数 mysqli_connect_errno()实现，该函数的说明如下：

```
int mysqli_connect_errno()
```

如果在连接 MySQL 数据库服务器时出现了错误，就会产生一个该错误对应的消息数字，利用该函数就可以获取该数字。下面为连接 MySQL 数据库时可能发生错误的错误号及错误内容：

```
1045：Access denied for user '用户名'@'用户密码'(user password:YES)
```

上述错误代表用户密码输入错误。

```
2005：Unknown MySQL server host 'MySQL 数据库服务器名'(11004)
```

上述错误代表 MySQL 数据库服务器名称输入错误。

```
2013：Loset connection to MySQL server during query
```

上述错误代表在执行查询时失去了与 MySQL 数据库的连接。

提示：在项目开发中，可以通过 try/catch 语句捕获通过 MySQLi 扩展库连接 MySQL 数据库所产生的异常。

设计过程

（1）建立 index.php 文件，用于建立 PHP 与 MySQL 数据库的连接，并输入错误的用户密码，代码如下：

```
$conn=mysqli_connect("localhost","root","root1");          //建立与 MySQL 数据库的连接
echo mysqli_connect_errno();                               //打印错误号
```

（2）建立 index_1.php 文件，用于建立 PHP 与 MySQL 数据库的连接，并输入错误的用户密码，代码如下：

```
$conn=mysqli_connect("localhost","root","root1");          //建立与 MySQL 数据库的连接
echo mysqli_connect_error();                               //打印错误提示
```



```
$sql=$conn->query("select * from tb_student,tb_score where tb_student.id=tb_score.sid");       //执行关联查询
$info=$sql->fetch_array(MYSQLI_ASSOC);                                                          //获得查询结果集
 if($info==NULL){                                                                               //未查到结果，将给出提示
      echo "暂无学生信息";
}else{
      do{
?>
<tr>
      <td height="20" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sno];?></div></td>
      <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[sname];?></div></td>
      <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[yw];?></div></td>
      <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[wy];?></div></td>
      <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[sx];?></div></td>
</tr>
<?php
      }while(   $info=$sql->fetch_array(MYSQLI_ASSOC));                                        //通过循环打印成绩信息
}
?>
```

秘笈心法

心法领悟 453：使用隐式连接查询实现多表之间的关联查询。

实现本实例所采用的多表之间的关联查询称为显示关联查询，查询语句如下：

```
select * from tb_student,tb_score where tb_student.id=tb_score.sid
```

除了使用上述显示关联查询外，还可以使用内连接查询方式实现多表之间的隐式关联查询，如本实例的多表连接查询可以更改成如下形式：

```
select * from tb_student inner join tb_score on tb_student.id=tb_score.sid
```

上述内连接查询中，使用 inner join 子句指定要连接的表名，使用 on 子句指定两个表的连接条件，如果还有其他查询条件，可以指定 where 子句。

说明： 为了规范数据库查询技术，隐式查询方式将成为多表连接查询的主流方式。

实例 454　通过 MySQLi 扩展技术实现内存的回收

光盘位置：光盘\MR\09\454

初级

趣味指数：★★★

实例说明

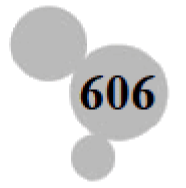
为了有效利用服务器的内存空间，在获取较大结果集时，一旦结束操作则有必要回收集合所需要的内存，由于 MySQLi 技术可以采用面向对象和面向过程两种编程方式，所以与之对应的内存回收方式也有两种，笔者将通过本实例讲解内存回收机制在项目开发过程中的意义和使用方法。运行本实例，如图 9.6 所示，图中为某班级学生主要科目成绩的列表，并在显示完该班成绩后对内存进行了回收。

学号	姓名	语文成绩	外语成绩	数学成绩
0312315	刘小华	88	60	94
0312316	金星星	60	85	76
0312317	黄小全	56	90	75
0312318	李小林	76	86	78

图 9.6　学生成绩列表

关键技术

PHP 为了迎合不同开发习惯的程序员，在 MySQLi 扩展技术中提供了面向对象和面向过程两种编程方式，所以实现内存回收也存在与之对应的两种方式。如果采用面向过程的编程方式，可以通过如下函数实现内存回收：



```
bool mysqli_close ( object link)
```

参数 link 指 mysqli_connect()函数返回的数据库连接句柄。

如果采用面向对象的编程方式，可以通过如下语句实现：

```
link->close()
```

link 指 PHP 预定义的 mysqli 类实例化后的对象。

设计过程

（1）建立 conn.php 文件，实现与 MySQL 数据库之间的连接。在 conn.php 文件中，首先使用 new 关键字对 mysqli 类进行实例化，然后通过实例后的对象调用 query()方法执行查询来设置数据库的字符编码为 UTF-8 编码，其实现代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");                    //建立与 MySQL 数据库的连接
$conn->query("set names utf8");                                                 //设置数据库字符编码
```

（2）通过 do...while 循环显示该班级所有学生的成绩。实现该功能时，首先使用 include_once 语句包含 conn.php 文件，然后通过调用 query()方法执行查询，并使用 fetch_array()方法获得查询结果集，最后通过 do...while 循环在页面中打印出学生的成绩信息，实现该过程的代码如下：

```
<?php
include_once("conn.php");                                                        //包含 conn.php 文件
$sql=$conn->query("select * from tb_student,tb_score where tb_student.id=tb_score.sid");   //执行查询
$info=$sql->fetch_array(MYSQLI_ASSOC);                                           //获得查询结果集
if($info==NULL){
      echo "暂无学生信息";
}else{
      do{                                                                        //通过循环打印查询结果
?>
<tr>
     <td height="20" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sno];?></div></td>
     <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[sname];?></div></td>
     <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[yw];?></div></td>
     <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[wy];?></div></td>
     <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[sx];?></div></td>
</tr>
<?php
       }while($info=$sql->fetch_array(MYSQLI_ASSOC));
}
?>
```

（3）实现内存的回收，这里分别调用执行 query()方法返回对象的 close()方法和 mysqli 实例后对象的 close()方法释放资源，代码如下：

```
$sql->close();
$conn->close();
```

秘笈心法

心法领悟 454：如何防止页面出现中文乱码。

编写 PHP 程序时，为了防止页面出现乱码，要求 PHP 文件编码、网页页面编码和数据库编码要一致，如果采用 UTF-8 编码，则上述三项内容都要设置为 UTF-8 编码，否则会出现乱码现象，那么如何对上述编码进行设置呢？设置 PHP 文件编码可以采用 UltraEdit、Edit Plus 和 Zend Studio 等编辑器，如使用 UltraEdit 设置 PHP 页面文件编码时，可以在另存页面文件时选择格式为 UTF-8，如图 9.7 所示。

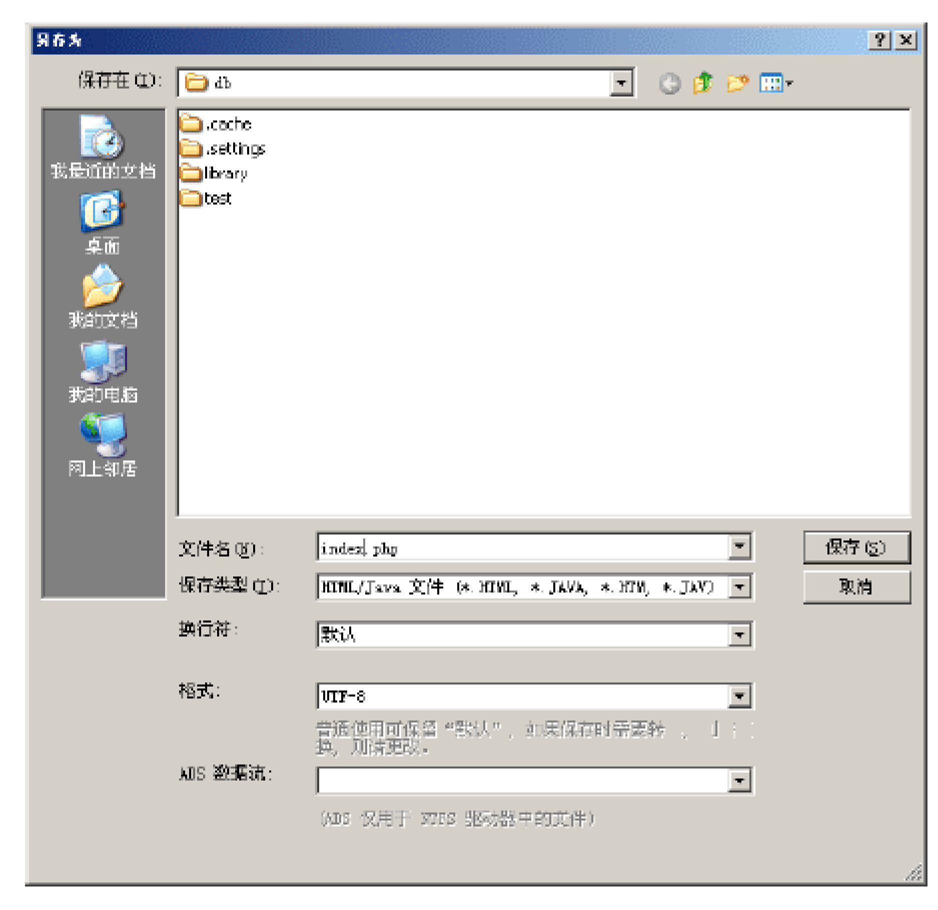


图 9.7　使用 UltraEdit 另存 PHP 文件为 UTF-8 编码格式

设置网页的页面编码可以使用 PHP 中的 header()函数或在 HTML 的 meta 标记中进行。如果页面完全由 PHP

（2）mysqli_fetch_row()函数根据所获取的记录生成数组，如果没有更多记录则返回 NULL，其语法如下：

```
mixed mysqli_fetch_row( mysqli_result result )
```

参数 result 为 mysqli_query()函数执行查询后的返回值。

设计过程

（1）建立 conn.php 文件，实现与 MySQL 数据库的连接。在 conn.php 文件中，首先使用 mysqli_connect()函数连接 db_database09 数据库，用户名是 root，密码为 111。然后通过 mysqli_connect_error()函数定义错误信息，最后通过 mysqli_query()函数设置数据库的编码格式，其代码如下：

```
<?php
$conn=mysqli_connect("localhost","root","111","db_database09");          //连接数据库
if (mysqli_connect_errno()) {
        printf("连接失败: %s\n", mysqli_connect_error());                //返回错误信息
        exit();
}
mysqli_query($conn,"set names utf8");                                    //设置数据库编码格式
?>
```

（2）新建 index.php 文件，创建 form 表单，将用户登录的用户名和密码提交到 index_ok.php 文件处理。

（3）创建 index_ok.php 文件，获取表单提交的用户名和密码，根据 mysqli_fetch_row()函数的返回值判断登录的用户名和密码是否正确，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                     //设置文件编码格式
include("conn/conn.php");                                                //包含数据库连接文件
if($_POST['user']!="" || $_POST['pass']!=""){                            //判断用户名和密码是否正确
     $sql=mysqli_query($conn,"select * from tb_login where username='".$_POST['user']."' and password='".$_POST['pass']."'");
     $info=mysqli_fetch_row($sql);                                       //获取查询结果
     if($info==NULL){
        echo "<script>alert('登录失败！'); window.location.href='index.php'</script>";
     }else{
        echo "<script>alert('登录成功！'); window.location.href='index.php';</script>";
     }
}
?>
```

秘笈心法

心法领悟 455：mysqli_fetch_row()函数的应用。

根据 mysqli_fetch_row()函数的返回值不但可以判断是否存在查询结果，如果与 while 语句结合使用，还可以输出查询结果中的数据。

实例 456　**用户注册**　光盘位置：光盘\MR\09\456　**初级**　趣味指数：★★★

实例说明

在本实例中运用 mysqli 函数库中的 mysqli_real_connect()函数实现与 MySQL 数据库的连接，从而实现用户注册的功能，并且介绍一种检测 SQL 语句中错误的方法。运行本实例，输入用户的注册信息，如果用户输入的用户名已经存在，则输出用户名已经存在的提示信息，否则提示用户注册成功，其运行结果如图 9.10 所示。



图 9.10　用户注册



```
}
mysqli_query($conn,"set names gb2312");                    //设置数据库编码格式
?>
```

（2）新建 index.php 文件，创建 form 表单，将用户注册信息提交到 index_ok.php 文件进行处理。

（3）创建 index_ok.php 文件，获取表单提交的用户注册信息，首先判断提交的用户名是否被占用，如果未被占用，则将用户注册信息添加到 tb_reg 表中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=gb2312" );                    //设置页面编码
include("conn/conn.php");                                                //包含数据库连接文件
if($_POST['user']!="" || $_POST['pwd1']!="" ){                           //判断输入的用户名和密码是否为空
        $check="select name from tb_reg where name='".$_POST['user']."'";  //定义验证用户名是否被占用的 SQL
        $result=mysqli_query($conn,$check);                              //执行查询语句
$info=mysqli_fetch_row($result);                                         //获取查询结果
      if($info!=NULL){
        echo "<script>alert('您输入的用户名已经存在！'); window.location.href='index.php';</script>";
      }else{
        $sql="insert into tb_reg(name,pwd,email,address)
values('".$_POST['user']."','".md5($_POST['pwd1'])."','".$_POST['email']."','".$_POST['address']."')";
        $result=mysqli_query($conn,$sql);
        echo mysqli_error($conn);                                        //返回 SQL 语句中的错误信息
        if($result){
                echo "<script>alert('注册成功!'); window.location.href='index.php';</script>";
        }else{
                        echo "<script>alert('注册失败!'); window.location.href='index.php';</script>";
        }
}
}
?>
```

秘笈心法

心法领悟 456：检测 SQL 语句中的错误。

在 mysqli 函数库中，可以通过 mysqli_error()函数来检测程序 SQL 语句中的错误，以帮助我们及时地发现 SQL 语句中的问题。

实例 457　数据的浏览

光盘位置：光盘\MR\09\457

初级　趣味指数：★★★

实例说明

在前面的实例中介绍了如何通过 mysqli 函数库中的函数实现用户注册和登录的功能，在本实例中将介绍如何通过 mysqli 函数库中的函数完成数据的浏览操作。运行本实例，将循环输出数据库中指定数据表的数据，其运行结果如图 9.11 所示。



图 9.11　浏览注册用户

关键技术

在本实例中，获取查询结果中数据应用的是 mysqli_fetch_object()函数，并结合 while 语句完成其返回对象中数据的输出。

mysqli_fetch_object()函数返回一个由执行查询后生成的表字段组成的对象，其语法如下：

```
mixed mysqli_fetch_object(object result)
```

参数 result 为 mysqli_query()函数执行查询的返回值。

获取返回对象中数据的格式如下：

```
<?php echo $obj->name; ?>
```

其中$obj 为返回的对象，name 为字段名称。

设计过程

（1）建立 conn.php 文件实现与 MySQL 数据库之间的连接。

（2）新建 index.php 文件，创建 form 表单，将用户登录信息提交到 index_ok.php 文件进行处理。

（3）创建 index_ok.php 文件，获取表单提交的用户登录信息，判断提交的用户名和密码是否正确，如果正确，则将登录用户名存储到 SESSION 变量中并跳转到 main.php 文件，其代码如下：

```
<?php
session_start();                                                    //初始化 SESSION 变量
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include("conn/conn.php");                                           //包含数据库连接文件
if($_POST['user']!="" || $_POST['pass']!=""){
    $sql=mysqli_query($conn,"select * from tb_login where username='".$_POST['user']."' and password='".$_POST['pass']."'");
    $info=mysqli_fetch_row($sql);
    if($info==NULL){
       echo "<script>alert('登录失败！'); window.location.href='index.php'</script>";
    }else{
$_SESSION['user']=$_POST['user'];
       echo "<script>alert('登录成功！'); window.location.href='main.php';</script>";
    }
}
?>
```

（4）创建 main.php 文件，连接数据库类执行查询语句，应用 mysqli_fetch_object()函数和 while 语句完成数据库中数据的循环输出，其关键代码如下：

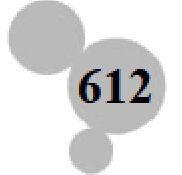
```
<?php
session_start();
include("conn/conn.php");
if($_SESSION['user']==true){
?>
//省略部分代码
<?php
       $sql=mysqli_query($conn,"select name,email,address from tb_reg ");
while($obj=mysqli_fetch_object($sql)){
?>
<tr>
           <td height="22" align="center"><?php echo $obj->name;?></td>
            <td align="center"><?php echo substr($obj->email,0,20);?></td>
            <td align="center"><?php echo $obj->address;?></td>
      </tr>
<?php
        }
?>
//省略部分代码
<?php
}else{
      echo "<script>alert('您不具备访问权限！');window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 457：返回查询结果集中的数据。

返回查询结果集中的数据，不但可以使用 mysqli_fetch_object()函数，还可以使用 mysqli_fetch_array()、mysqli_fetch_assoc()和 mysqli_fetch_row()函数。

通过 mysqli_fetch_array()函数返回的查询结果集可以使用数字索引或者关联索引来读取数据；而通过 mysqli_fetch_assoc()函数返回的查询结果集必须使用关联索引来获取数据，通过 mysqli_fetch_row()函数返回的查询结果集必须使用数字索引来获取数据。



实例 458　数据库、数据表中数据的动态输出

光盘位置：光盘\MR\09\458

初级

趣味指数：★★★

实例说明

在本实例中应用 mysqli 函数库中的函数动态地对数据库和数据表进行操作，统计数据表中的字段和记录数，并且循环输出数据表中的数据，其运行结果如图 9.12 所示。



图 9.12　浏览注册用户

关键技术

（1）统计查询结果中的字段数应用的是 mysqli_num_fields()函数，其语法如下：

```
int mysqli_num_fields(mysqli_result result)
```

参数 result 为 mysqli_query()、mysqli_store_result()或 mysqli_use_result()函数执行查询后的返回值。

（2）统计查询结果中的记录数应用的是 mysqli_num_rows()函数，其语法如下：

```
int mysqli_num_rows(mysqli_result result)
```

参数 result 为 mysqli_query()、mysqli_store_result()或 mysqli_use_result()函数执行查询后的返回值。

（3）获取数据表中字段信息应用的是 mysqli_fetch_fields()函数。它返回一个对象数组，该数组包含某表字段的信息，如果该表的字段为空，则返回 FALSE，其语法如下：

```
array mysqli_fetch_fields(mysqli_result result)
```

参数 result 为 mysqli_query()函数的返回值，返回的对象数组可以调用的属性如表 9.5 所示。

表 9.5　mysqli_fetch_fields()函数返回的对象数组可以调用的属性说明

属　性	说　明
name	返回字段的名称
orgname	返回字段的原始名称
table	返回字段所属的表名
orgtable	返回字段所属的原始表名
def	返回该字段的默认值
max_length	返回字段的最大宽度
length	字段的宽度
charsetnr	返回字段所属的字符集
flags	返回一个描述字段的位标记
type	返回字段的数据类型
decimals	返回字段小数部分的位数

秘笈心法

心法领悟 458：动态读取数据库的优势。

动态读取数据库可以增加页面的灵活性，不必拘泥于某个数据库、数据表的格式，此方法更适合于在后台管理系统中对数据进行管理。

9.2　MySQL 视图的应用

视图也称虚表，包括执行某个查询返回的一组记录。视图可以简化获取数据的方法，并通过别名来实现抽象查询。

实例 459	在 MySQL 数据库中创建视图 光盘位置：光盘\MR\09\459	初级 趣味指数：★★★

实例说明

为了简化查询、提高系统安全性，在项目开发过程中，通常采用为数据库中的某个表建立视图的方式。在建立的视图中只有开发人员所关心的字段。运行本实例，如图 9.13 所示，分别在图中输入登录用户的用户名和密码，单击“进入”按钮后，如果用户输入的用户名或密码错误则给出错误提示，反之，如果输入的用户名和密码正确，则提示成功登录。由于本实例采用视图建立了虚表，所以应首先在命令提示符窗口中建立视图，如图 9.14 所示。



图 9.13　用户身份验证

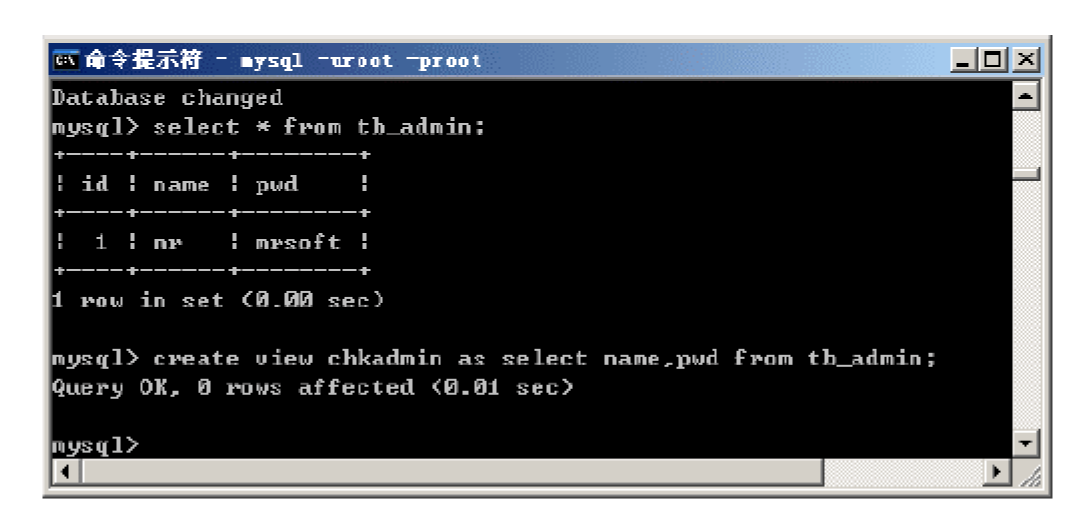


图 9.14　创建视图

关键技术

在 MySQL 中创建视图可以通过 create view 语句来实现，具体创建格式如下：

```
create [ or deplace] [algorithm={merge | temptable | undefined}] view view_name [( column_list)] as select_statement [with [cascaded | local] check option]
```

参数说明：

view_name：新建视图的名称。

select_statement：SQL 查询语句，用于限定虚表的内容。

algorithm={merge | temptable | undefined}属性用于优化 MySQL 视图的执行，该属性有 3 个可用的设置。下面将介绍这 3 个设置的使用方法：

merge：该参数将 MySQL 执行视图时传入的任何子句合并到视图的查询定义中。

temptable：如果视图低层表中的数据有变化，将在下次通过表时立即反映出来。

undefined：当查询结果和视图结果为一一对应关系时，MySQL 将 algorithm 设定为 temptable。

设计过程

（1）建立数据库及数据库表。本实例中将数据表命名为 tb_admin，其创建过程可以通过 phpMyAdmin 实现。

（2）完成数据表的创建后，在命令提示符下建立视图，创建视图的代码如下：

```
create view chkadmin as select name, pwd from tb_admin
```

创建的视图 chkadmin 中只包含 name 和 pwd 两个字段信息，这样使得程序的其他字段信息不可见，从而可以有效提高程序的安全性。

（3）采用 PHP 中的 MySQL 函数库建立与 MySQL 数据库的连接。首先使用 mysql_connect()函数连接 MySQL 数据库，然后使用函数 mysql_select_db()选择数据库 db_database09，最后使用函数 mysql_query()执行 set names utf8 语句来设置数据库的编码为 UTF-8 编码，实现该过程的代码如下：

```
$conn=mysql_connect("localhost","root","111");          //建立与 MySQL 数据库的连接
mysql_select_db("db_database09",$conn);                 //选择数据库
mysql_query("set names utf8");                          //设置数据库字符编码
```

（4）判断用户是否单击了“进入”按钮，如果是，则通过视图 chkadmin 对用户身份进行验证，该过程的代码如下：

```
if($_POST[submit]!=""){
    include_once("conn.php");                           //包含 conn.php 文件
    $name=$_POST[name];                                 //接收提交的用户名
    $pwd=$_POST[pwd];                                   //接收提交的用户密码
    $sql=mysql_query("select * from chkadmin where name='".$name."' and
pwd='".$pwd."'",$conn);                                 //执行查询
    $info=mysql_fetch_array($sql);                      //返回结果集
    if($info==false){                                   //判断用户名和密码是否正确
        echo "<script>alert('用户名或密码输入错误!');history.back();</script>";
        exit;
    }else{
        echo "<br><div align=center>登录成功!</div>";   //如果正确则提示登录成功
    }
}
```

秘笈心法

心法领悟 459：使用视图的实际意义。

视图在项目开发过程中具有非常重要的实际意义，主要表现在如下两点：

（1）使用视图可以使查询更加简化。在程序编写中，如果所要执行的查询较为复杂，可以将其转化为视图，从而简化查询。

（2）使用视图还可以提高程序的安全性。如果只想让用户访问某表中的某几个指定的字段，可以创建一个只包含这些字段的视图，而不用将整个表的所有字段的访问权限都分配给该用户。

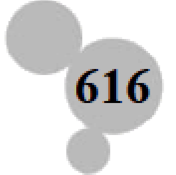
技巧： 虽然不使用视图也可以完成对数据库的操作，但在字段数量较多或安全性要求较高的模块中建议通过视图完成查询，从而有效地提高程序的可维护性和安全性。

实例 460　修改 MySQL 数据库中的视图　　初级

光盘位置：光盘\MR\09\460　　趣味指数：★★★

实例说明

通过本实例主要讲解如何对已经创建完成的 MySQL 视图进行修改，在制作本实例时，首先需要在数据库中创建数据表 tb_changeview，其结构如图 9.15 所示，然后创建视图 userinfo，该视图中包含 tb_changeview 表中的所有字段，完成上述操作后，使用 alter view 语句对视图进行修改。



	字段	类型	整理	属性	Null	默认	额外	操作
☐	id	int(8)			否		auto_increment	
☐	username	varchar(50)	gb2312_chinese_ci		否			
☐	userpwd	varchar(50)	gb2312_chinese_ci		否			

图 9.15　创建表 changeview

关键技术

在 MySQL 中修改视图可以通过 alter view 语句实现，alter view 语句的具体使用说明如下：

```
alter view [algorithm={merge | temptable | undefined} ]view view_name [(column_list)] as select_statement[with [cascaded | local] check option]
```

参数说明：

algorithm：该参数已经在创建视图中作了介绍，这里不再赘述。

view_name：视图的名称。

select_statement：SQL 语句，用于限定视图。

设计过程

（1）使用 phpMyAdmin 创建数据库及数据库表，本实例中将数据表命名为 tb_changeview。

（2）使用 create view 语句创建视图 userinfo，其创建代码如下：

```
create view userinfo as select * from tb_changeview
```

（3）使用 select 语句查询视图 userinfo，查询代码如下：

```
select * from userinfo
```

查看执行结果，可以发现 tb_changeview 表中的所有字段都一一列出，如图 9.16 所示。

（4）下面开始修改视图，要求修改后只列出 username 字段，代码如下：

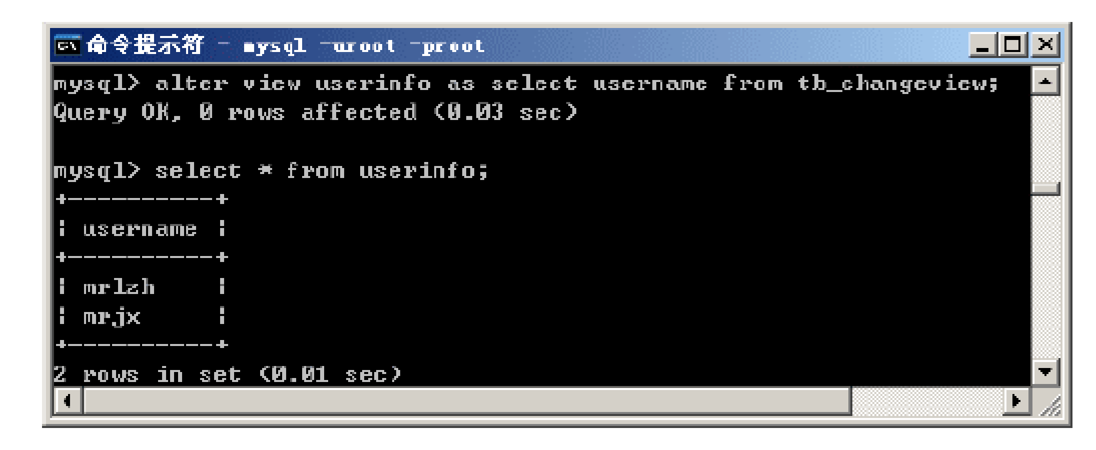
```
alter view userinfo as select * from tb_changeview
```

（5）重新查询视图 userinfo，从结果中可以发现，此时只列出 username 一个字段，从而说明视图修改成功，如图 9.17 所示。

```
命令提示符 - mysql -uroot -proot
mysql> create view userinfo as select * from tb_changeview;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from userinfo;
+----+----------+----------+
| id | username | userpwd  |
+----+----------+----------+
|  1 | mrlzh    | 12345678 |
|  2 | mrjx     | 87654321 |
+----+----------+----------+
2 rows in set (0.01 sec)
```

图 9.16　创建新视图



```
命令提示符 - mysql -uroot -proot
mysql> alter view userinfo as select username from tb_changeview;
Query OK, 0 rows affected (0.03 sec)

mysql> select * from userinfo;
+----------+
| username |
+----------+
| mrlzh    |
| mrjx     |
+----------+
2 rows in set (0.01 sec)
```

图 9.17　修改视图

注意：如果数据库采用 UTF-8 编码，在命令提示符下通过 select 语句显示查询结果时会出现中文乱码现象。

秘笈心法

心法领悟 460：在 phpMyAdmin 的查询窗口中创建和修改视图。

在实现本实例时，创建视图的过程是在 Windows 系统的命令提示符窗口中实现的，为了提高开发效率、简化程序开发人员的操作，MySQL 数据库的创建和维护过程一般在 MySQL 的图形管理工具 phpMyAdmin 中进行，phpMyAdmin 为 MySQL 数据库提供了较为完善的创建、维护及备份等操作。同样，如果使用 phpMyAdmin 的查询窗口也可以执行 SQL 命令，所以可以在该查询窗口中创建视图，其具体创建过程如下：

（1）打开 phpMyAdmin 管理页面并选择要管理的数据库，单击如图 9.18 所示的“查询窗口”按钮打开 phpMyAdmin 的查询窗口。

（2）phpMyAdmin 的查询窗口如图 9.19 所示，可以在该窗口中输入 SQL 命令实现对指定数据库及表的查询操作。例如在该查询窗口中输入创建视图的 SQL 命令，然后单击图中的“执行”按钮即可成功地实现视图的创建操作。

秘笈心法

心法领悟 461：在 phpMyAdmin 的命令窗口中删除视图。

要删除指定的视图，可以在 phpMyAdmin 的命令窗口中实现。首先打开 phpMyAdmin 的命令窗口，如图 9.22 所示，然后在该窗口中输入 drop view 语句，最后单击“执行”按钮即可成功删除指定的视图。

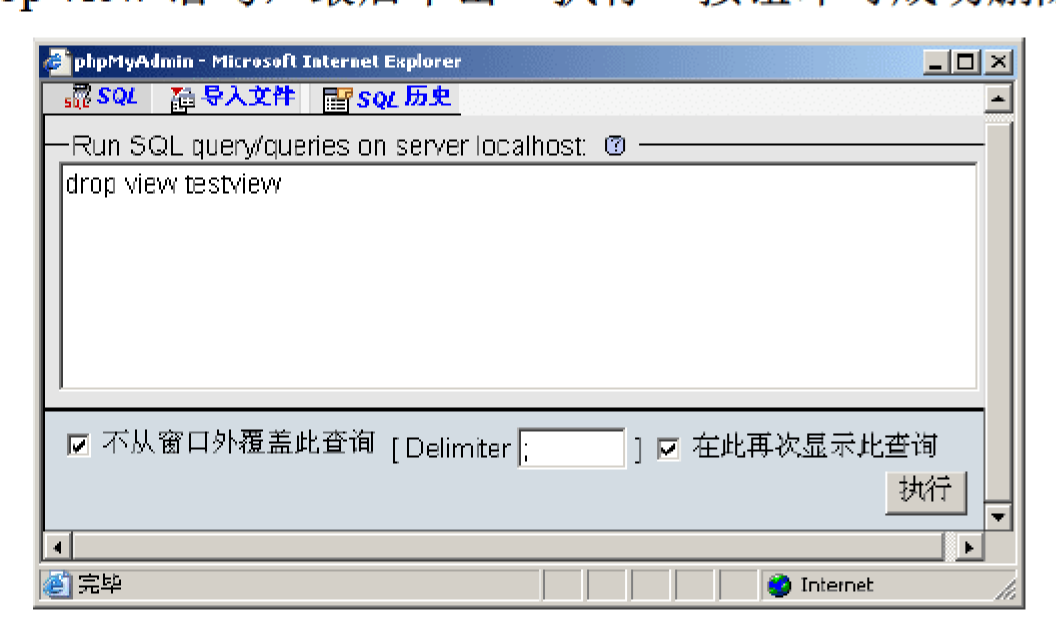


图 9.22　在 phpMyAdmin 的命令窗口中删除视图

实例 462　使用 MySQL 视图查询学生成绩信息　初级

光盘位置：光盘\MR\09\462　趣味指数：★★★

实例说明

在实际项目开发过程中数据表中可能有很多字段，但某个模块可能只需要其中的几个。为了提高查询速度和简化操作，可以将该模块需要的字段单独提取出来放在某个视图中，例如本实例涉及学生表和成绩表，在建立的视图中只含有与学生成绩有关的字段，如图 9.23 所示。

运行本实例，如图 9.24 所示，图中查询结果显示的内容即为视图中所有字段中的内容。

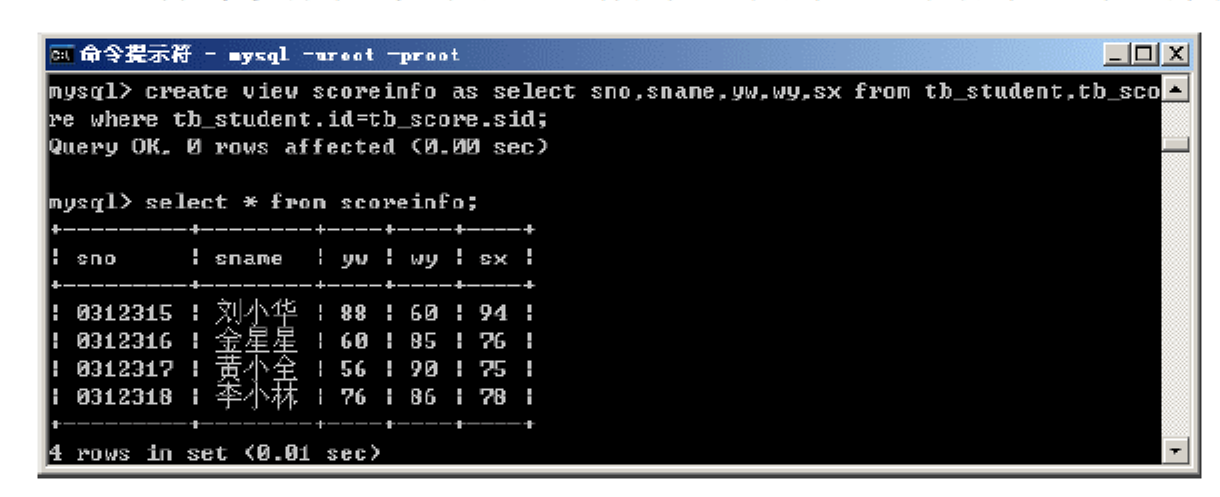


图 9.23　创建视图



图 9.24　学生成绩列表

关键技术

本实例创建的视图涉及多表查询，这说明多个表之间可以通过视图来组合为一个整体，这样对视图的操作就相当于多表查询。但如果这些表中有相同的字段，必须按如下方式书写重名的字段：

```
表名 1.字段名,表名 2.字段名...
```

设计过程

（1）创建视图 scoreinfo，通过该视图显示学生成绩信息，该视图的创建代码如下：

```
create view scoreinfo as select sno,sname,yw,wy,sx from tb_student,tb_score where tb_student.id=tb_score.sid
```

（2）建立数据库连接文件 conn.php，实现与 MySQL 数据库的连接，并设置数据库字符集为 UTF-8 编码，代码如下：




图 9.25　用户登录窗口

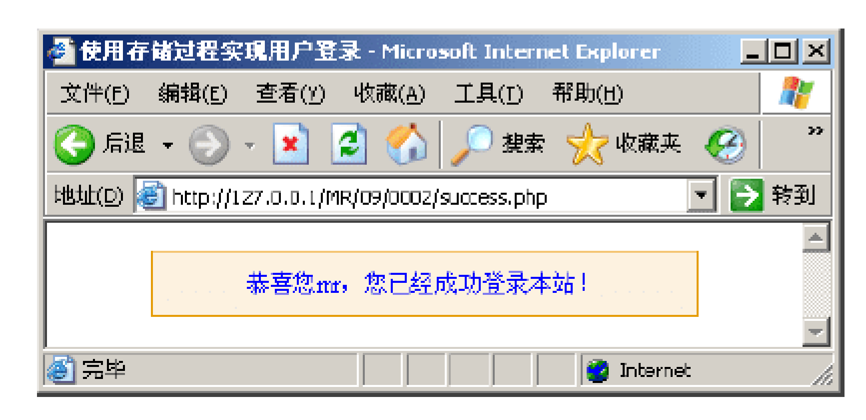


图 9.26　登录成功的提示信息

关键技术

在创建本实例的存储过程中，为了防止结束符冲突，首先使用 delimiter 语句将 MySQL 默认的结束符“;”更改为“//”，然后创建存储过程，其创建代码如图 9.27 所示。

```
命令提示符 - mysql -uroot -proot
C:\>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.0.51b-community-nt-log MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use db_database09;
Database changed
mysql> delimiter //
mysql> create procedure pro_login(in un varchar(50), in pwd varchar(50))
    -> begin
    -> select * from tb_login where username=un and password = pwd;
    -> end;
    -> //
Query OK, 0 rows affected (0.06 sec)
```

图 9.27　创建用户登录验证的存储过程

创建完成上述存储过程后，在程序中使用 call 语句调用要执行的存储过程，本实例使用 MySQLi 扩展库执行存储过程，其实现代码如下：

```
$sql = $mysqli->query("call pro_login('".$username."', '".$password."')");
```

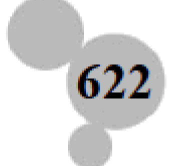
设计过程

（1）通过 MySQLi 扩展库建立与 MySQL 数据库的连接，并设置数据库字符集为 UTF-8，实现代码如下：

```
$mysqli=new mysqli("localhost","root","111","db_database09");          //建立与 MySQL 数据库的连接
$mysqli->query("set names utf8");                                       //设置数据库字符编码为 UTF-8
```

（2）建立用户登录表单，当用户在表单中录入用户名和密码，并单击“提交”按钮后，通过如下代码验证用户的登录信息是否正确：

```
if(isset($_POST['username']) && trim($_POST['username'])!='')          //判断用户是否提交了表单
{
    require_once 'Db.php';                                              //包含 Db.php 文件
    $username = trim($_POST['username']);                               //获得提交的用户名
    $password = trim($_POST['password']);                               //获得提交的用户密码
    $sql = $mysqli->query("call pro_login('".$username."', '".$password."')");   //调用存储过程，执行查询
    $info = $sql->fetch_array(MYSQLI_ASSOC);                            //获得查询第一条记录
    if($info != null){                                                  //判断是否为 NULL 值
        $_SESSION['loginUsername'] = $username;                         //将用户名保存到 SESSION 中
        echo '<script>window.location.href="success.php";</script>';   //重定向到 success.php 页面
    }else {
        echo '<div style="width:300px; height:30px; line-height:30px; border:1px solid #E59B04; background-color:#FCF2E0; color:#FF0000;">用户
```



设计过程

（1）通过 PHP 预定义类 mysqli 实现与 MySQL 数据库的连接，代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");
$conn->query("set names utf8");
```

（2）调用存储过程 pro_reg 实现将用户录入的注册信息保存到数据库，代码如下：

```
if($sql=$conn->query("call pro_reg('".$nc."','".$pwd."','".$email."','".$address."')")){
    echo "<script>alert('用户注册成功!');</script>";
}else{
     echo "<script>alert('用户注册失败!');</script>";
}
```

秘笈心法

心法领悟 465：使用 PDO 数据库抽象类执行存储过程。

实现本实例时，使用 MySQLi 扩展库执行存储过程，除使用该方法外，还可以使用 PDO 数据库抽象类执行存储过程，在实现本实例时，可以将保存用户注册信息的代码进行如下更改，来演示使用 PDO 技术执行 MySQL 存储过程的方法。

```
$db = new PDO('mysql:host=localhost;dbname=db_database09', 'root', 'root');      //实例 PDO 类，连接 MySQL 数据库
$stmt = $db->prepare("call pro_reg(?, ?, ?, ?)");                                 //建立查询的预处理句柄
$stmt->bindParam(1, $_POST['nc']);                                                //绑定参数
$stmt->bindParam(2, $_POST['pwd']);
$stmt->bindParam(3, $_POST['email']);
$stmt->bindParam(4, $_POST['address']);
$stmt->execute();                                                                 //执行查询
```

说明：使用 PDO 连接 MySQL 数据库时，连接字符串 mysql:host=localhost;dbname=db_database09 中内容的顺序可以颠倒。

9.4 MySQL 触发器

触发器在数据库系统开发过程中具有非常重要的作用，例如可以防止有害数据录入数据库、可以改变或取消 insert、update 和 delete 语句的执行及在一个会话中监听数据库中的数据的改变。

实例 466	创建 MySQL 触发器 光盘位置：光盘\MR\09\466	初级 趣味指数：★★★

实例说明

如果用户打算在数据库中通过触发器实现某一动作的监听，那么首先应该创建触发器。触发器是在命令提示符下创建的，如图 9.31 所示。

```
命令提示符 - mysql -uroot -proot
mysql> use db_database09;
Database changed
mysql> create table tb_test(t_id varchar(20),t_name varchar(20));
Query OK, 0 rows affected (0.08 sec)

mysql> select * from tb_test;
Empty set (0.01 sec)
mysql> delimiter //
mysql> create trigger test_tri
    -> before insert on tb_test
    -> for each row
    -> set new.t_name='mrsoft'//
Query OK, 0 rows affected (0.02 sec)
mysql> insert  into tb_test(t_id,t_name)values('mr0001','123')//
Query OK, 1 row affected (0.01 sec)

mysql> select * from tb_test//
+--------+--------+
| t_id   | t_name |
+--------+--------+
| mr0001 | mrsoft |
+--------+--------+
1 row in set (0.00 sec)
```

图 9.31　触发器创建过程

关键技术

MySQL 数据库中创建触发器的格式如下：

```
create trigger <触发器名称>
{ before | after}
{insert | update | delete}
on <表名>
for each row
<触发器 SQL 语句>
```

参数说明：

create trigger <触发器名称>：创建一个新触发器，并指定触发器的名称。

{ before | after}：用于指定在 insert、update 或 delete 语句执行前触发还是在语句执行后触发。

on <表名>：用于指定响应该触发器的表名。

for each row：触发器的执行间隔。for each row 通知触发器每隔一行执行一次动作，而不是对整个表执行一次。

<触发器 SQL 语句>：触发器要执行的 SQL 语句。

设计过程

（1）在命令提示符下创建数据表 tb_test，代码如下：

```
create table tb_test(t_id varchar(20),t_name varchar(20))
```

（2）将换行标记转换为“//”，代码如下：

```
delimiter //
```

（3）创建触发器，实现无论用户向表 tb_test 添加什么数据都使字段 t_name 的内容为 mrsoft，代码如下：

```
create trigger test_tri
before insert on tb_test
for each row
set new.t_name='mrsoft'
```

（4）向表 tb_test 中添加一条记录，并查看结果，代码如下：

```
insert into tb_test(t_id,t_name) values('mr0001', '123')//
select * from tb_test
```

秘笈心法

心法领悟 466：触发器中执行多条 SQL 语句。

如果该触发器要执行多条 SQL 语句，要将多条语句放在 begin...end 块中。

实例 467　**查看 MySQL 触发器**　初级

光盘位置：光盘\MR\09\467　　趣味指数：★★★

实例说明

在 MySQL 数据库中创建触发器前应先查看数据库中的触发器，这样既可以使开发人员对指定的数据库中的所有触发器及其功能有一个直观的把握，而且可以避免创建同名或类似功能的触发器。运行本实例，如图 9.32 所示，首先在图中的文本框中输入要查看触发器的数据库名称，然后单击“查看”按钮，即可将该数据库中所有触发器的详细信息显示出来。



图 9.32　查看触发器

关键技术

查看 MySQL 数据库中触发器的详细信息，可以通过如下语句实现：

删除相关 id 的工作后，还没来得及删除学生信息表中该学生的信息时就发生停电等意外，则再重新查找该学生的成绩时是无法查找到的。

本实例将采用事务处理方式，对学生信息表和学生成绩表中的数据进行删除。运行本实例，学生信息及学生成绩信息分别如图 9.33 和图 9.34 所示。当删除图 9.33 中的学生信息后，查看学生成绩信息时可以发现，与该学生对应的成绩也被全部删除。



图 9.33　查看学生信息



图 9.34　查看学生成绩

■ 关键技术

事务的处理可以通过 PHP 的预定义类 mysqli 的以下方法实现。

（1）autocommit(boolean)：该方法用于限定查询结果是否自动提交，如果参数为 true 则自动提交，如果参数为 false 则关闭自动提交。MySQL 数据库默认为自动提交。

（2）rollback()：利用 mysqli 类中的该方法可以实现事务的回滚。

（3）commit()：利用该方法可以实现提交所有查询。

■ 设计过程

（1）建立数据库及数据表并实现与数据的连接，代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");
$conn->query("set names utf8");
```

（2）显示所有学生的基本信息，代码如下：

```
<?php
include_once("conn.php");
$sql=$conn->query("select * from tb_stu");
$info=$sql->fetch_array(MYSQLI_ASSOC);
if($info==NULL){
      echo "暂无学生信息!";
}else{
      do{
?>
      <tr>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sname];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sno];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sage];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[saddress];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[ssfzh];?></div></td>
            <td bgcolor="#FFFFFF"><div align="center"><a href="javascript:if(window.confirm('确定删除该学生信息
么?')==true){window.location.href='delete.php?id=<?php echo $info[id];?>';}">删除</a></div></td>
      </tr>
<?php
      }while($info=$sql->fetch_array(MYSQLI_ASSOC));
}
?>
```

在实现该模块功能时，可以利用 JavaScript 实现该页与 delete.php 页面的信息传递，代码如下：

```
<a href="javascript:if(window.confirm('确定删除该学生信息么?')==true){window.location.href='delete.php?id=<?php echo $info[id];?>';}">删除</a>
```

window 对象的 confirm()方法用于弹出一个对话框，提示用户是否真正删除某学生的所有信息。

（3）利用事务处理机制对学生的基本信息进行删除，代码如下：

```
$id=$_GET[id];
include_once("conn.php");
$conn->autocommit(false);
if(!$conn->query("delete from tb_sco where id='".$id."'")){
```

参数说明：

string query：SQL 语句。

link_identifier：连接标识。

（3）mssql_fetch_row()函数取得一行数据，保存于数组中，其语法如下：

```
array mssql_fetch_row ( resource result )
```

参数说明：

resource result：结果集。

本实例与 SQL Server 建立连接的代码如下：

```
$conn = mssql_connect("localhost","sa","") or die ("Connect MySQL False");
mssql_select_db("db_database10",$conn);
```

设计过程

在本实例中，通过 mssql_connect()函数连接 SQL Server，应用 mssql_query()函数执行查询语句，应用 mssql_fetch_row()函数和 while 语句完成用户信息的循环输出，其关键代码如下：

```
<?php
        $conn = mssql_connect ( "localhost", "sa", "" ) or die ( "Connect SQL Server False" );          //连接 SQL Server 服务器
        mssql_select_db ( "db_database10", $conn );                                                     //连接 db_database10 数据库
        if ($_GET [id] == '1') {
?>
<?php
                if($rs = mssql_query("select * from tb_demo01")){                                       //如果返回结果集
                        while($rst = mssql_fetch_row($rs)){                                             //循环输出结果集
?>
                <tr>
                        <td bgcolor="#FFFFFF" align="center"><?php echo $rst[0];?></td>
                        <td bgcolor="#FFFFFF" align="center"><?php echo $rst[1];?></td>
                </tr>
<?php
                        }
                }else{
                        echo "<script>alert('数据表存在问题');</script>";
                }
        }
?>
```

秘笈心法

心法领悟 471：如何检测 PHP 是否支持 SQL Server 数据库。

编写脚本文件，输入如下代码：

```
<?php
    echo phpinfo();
?>
```

经 IE 浏览器运行本文件，如果网页中显示 mssql 内容，表示 PHP 支持 SQL Server 数据库。

如果用户使用的 SQL Server 数据库是 2000 版本，可能会出现 unable connect server:localhost 错误，造成这种错误的原因一般分为两种情况：

（1）代码存在错误。检查 SQL Server 服务管理器是否处于正在运行的状态。

（2）系统可能缺少 ntwdblib.dll 文件。用户可以手动下载 ntwdblib.dll-2000.80.2039.0 文件，将此文件复制到 Apache\bin 文件夹和 php\bin 文件夹下，然后重启 SQL Server 服务器和 Apache 服务器即可。也可为 SQL Server 打上 sp4 补丁，然后重启服务器。

10.2　操作数据库

SQL Server 数据库与 MySQL 数据库虽然都是存储信息的载体，但是在操作方法与函数上略有不同。本节

将具体讲述对 SQL Server 数据库的相关操作。

实例 472 通过 ADO 方式实现指定时间段的信息检索

光盘位置：光盘\MR\10\472

高级

趣味指数：★★★★

实例说明

在进行网站开发过程中，经常用到对指定时间段的信息检索，例如，检索指定时间段内有哪些 PHP 类图书出版等。运行本实例，当单击“信息检索”超链接时，将显示 2009—2010 年明日科技编写的所有 PHP 类图书，其运行效果如图 10.3 所示。

通过ADO方式实现指定时间段的信息检索

信息检索

图书ID	图书名称	图书价格	出版社
1	《PHP范例宝典》	78	人民邮电出版社
2	《PHP实战宝典》	74	清华大学出版社
3	《PHP网络编程》	65	东财大学出版社

图 10.3　通过 ADO 实现指定时间段的信息检索

关键技术

本实例通过 ADO 操作 SQL Server 数据库，执行指定时间段内的信息查询操作，其信息检索的 SQL 语句代码如下：

```
$sql = "select id,bname,price,pub from tb_demo03 where date>='2009-01-01' and date<='2010-12-31'";
```

设计过程

创建 index.php 文件，首先通过 ADO 连接 SQL Server 数据库，然后建立记录集，实现用户信息的循环输出，其关键代码如下：

```
<?php
$conn = new com ( "adodb.connection" );
$connstr = "provider = sqloledb;data source=PC-201006101638;uid=sa;pwd=;database=db_database10";        //ADO 连接数据库
$conn->open ( $connstr );
if ($_GET [id] == "1") {
        $sql = "select id,bname,price,pub from tb_demo03 where date>='2009-01-01' and date<='2010-12-31'";
        $rs = new com ( "adodb.recordset" );
        $rs->open ( $sql, $conn );
        if ($rs->eof || $rs->bof) {
                echo "数据库暂无信息";
        } else {
                ?>
        <tr>
                <td align="center">图书 ID</td>
                <td align="center">图书名称</td>
                <td align="center">图书价格</td>
                <td align="center">出版社</td>
        </tr>
<?php
                while ( ! $rs->eof ) {
                        ?>
        <tr>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( id )->value ); ?></td>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( bname )->value ); ?></td>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( price )->value ); ?></td>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( pub )->value ); ?></td>
        </tr>
```

```
<?php
                $rs -> movenext;
                }
        }
}
?>
```

秘笈心法

心法领悟 472：解决页面乱码问题。

由于在 SQL Server 中使用的编码格式是 GBK，而页面中的编码格式为 UFT-8，所以需使用转码函数 iconv()对输出结果进行转码，否则将输出乱码。

实例 473	检索商品信息 光盘位置：光盘\MR\10\473	高级 趣味指数：★★★★

实例说明

在实际开发中，很多程序都会涉及对信息的检索。运行本实例，输入商品 ID，单击“检索”按钮，将查找出匹配的商品信息并且对商品名称描红显示，运行结果如图 10.4 所示。



图 10.4　商品信息的检索

关键技术

本实例通过 mssql_connect()函数连接 SQL Server 服务器，通过 mssql_select_db()函数连接 db_database10 数据库，通过 mssql_query()函数执行查询语句，通过 mssql_fetch_row()函数获取查询结果中的数据，其关键代码如下：

```
<?php
$conn = mssql_connect ( "localhost", "sa", "" ) or die ( "Connect SQL Server False" );          //连接数据库服务器
mssql_select_db ( "db_database10", $conn ) or die ( "Connect Database False" );                 //连接数据库
if ($_POST [sub]) {                                                                             //判断提交按钮是否为空
        if ($_POST [text] == "" || $_POST [text] == "输入商品 ID") {                             //如果提交 ID 值为空，则给出提示信息
                echo "<script>alert('请输入查询关键字');</script>";
        } else {
                $rs = mssql_query ( "select * from tb_demo04 where id=" . $_POST [text] );      //执行查询语句
?>
```

设计过程

（1）在 index.php 文件中首先创建 form 表单，将商品的 ID 值提交到本页，其代码如下：

```
<form action="" method="post">
        <input class="one" type="text" name="text" value="输入商品 ID" size="15" onfocus="this.value=''" />
        <input class="two" type="submit" name="sub" value=" " />
</form>
```

（2）在 index.php 文件中，通过 mssql_connect()函数连接 SQL Server，根据 form 表单中提交的 ID 值执行查询语句并循环输出查询结果，代码如下：

```
<?php
$conn = mssql_connect ( "localhost", "sa", "" ) or die ( "Connect SQL Server False" );
```

内部的行指针移动到指定的行号。如果成功则返回 TRUE，否则返回 FALSE，其语法如下：

```
bool mssql_data_seek( resource result_identifier, int row_number )
```

参数说明：

result_identifier：返回类型为 resource 的结果集，该结果集从 mssql_query()函数的调用中得到。

row_number：想要设定的新的结果集指针行数。

设计过程

（1）新建 conn.php 文件，通过 mssql 函数库连接 SQL Server 数据库。

（2）新建 index.php 文件，首先创建 form 表单，提交定位查询所使用的数字。然后对表单中提交的定位查询数据进行判断，如果提交的值正确，则应用 mssql_data_seek()函数定位指针位置，通过 mysql_fetch_row()函数获取查询结果中的记录集；否则给出提示信息，并且通过 while 语句和 mssql_fetch_row()函数循环输出数据库中的所有记录，其关键代码如下：

```
if ($_POST [text] == "" || $_POST [text] == "请输入数字") {
	echo "<script>alert('请在文本框中输入信息');</script>";
} else {
	if (! preg_match ( "/\d/", $_POST [text] )) {
		echo "<script>alert('请输入数字')</script>";
	} else {
		if (@mssql_data_seek ( $rs, $_POST [text] )) {
			$ts = mssql_fetch_row ( $rs );
			?>
	<tr>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts [0] )?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts [1] )?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts [2] )?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts [3] )?></td>
	</tr>
<?php
		} else {
			echo "<script>alert('输入无效数据或超出范围');</script>";
		}
	}
}
```

秘笈心法

心法领悟 474：使用 mssql_data_seek()函数的注意事项。

使用此函数时要注意，参数 row_number 是从 0 开始的，所以若在本实例文本框中输入数字 1，获取的是数据库中的第 2 条数据。

实例 475	取得数据中指定的字段信息 光盘位置：光盘\MR\10\475	高级 趣味指数：★★★★

实例说明

在 PHP 操作 MySQL 数据库中，获取数据字段信息使用 mysql_fetch_field ()函数，而在 mssql 函数库中操作 SQL Server 数据库获取数据字段信息使用 mssql_fetch_field()函数。在本实例中，通过 mssql_fetch_field()函数获取数据表中指定字段的信息，在图 10.7 所示的文本框中输入一个数字值，单击“确定”按钮，将输出如图 10.8 所示的内容。

```
                <td class="o">column_source</td>
                <td class="o">type</td>
                <td class="o">max_length</td>
        </tr>
        <tr>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->name )?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->column_source )?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->type )?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->max_length )?></td>
        </tr>
<?php
        }
}
} else {
?>
```

秘笈心法

心法领悟 475：使用 mssql_fetch_field()函数的注意事项。

mssql_fetch_field()函数可以用来从某个查询结果中获取字段的信息。如果没有指定字段偏移量，则下一个尚未被 mssql_fetch_field()函数获取的字段将被提取。

实例 476　利用 mssql_field_name()函数获取字段名称

光盘位置：光盘\MR\10\476

高级

趣味指数：★★★★

实例说明

实例 475 中通过 mssql_fetch_field()函数返回对象的 name 属性获取字段的名称，下面将介绍在 mssql 函数库中专门用来获取字段名称的方法。运行本实例，在文本框中输入一个有效的字段值，单击“确定”按钮，将输出如图 10.9 所示的页面。

利用mssql_field_name()获取字段名称

2　确定

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00
数据表字段名称为	price		

图 10.9　利用 mssql_field_name()函数获取字段名称

关键技术

本实例主要应用 mssql_field_name()函数获取字段的名称，其语法如下：

```
string mssql_field_name ( resource result, int field_index )
```

该函数返回指定字段索引的字段名。参数 result 必须是一个合法的结果标识符，参数 field_index 是该字段的数字偏移量。

设计过程

（1）新建 conn.php 文件，通过 mssql 函数库连接 SQL Server 数据库。

（2）新建 index.php 文件，首先创建 form 表单，提交定位查询所使用的数字。然后执行查询语句，并通过 while 语句和 mssql_fetch_row()函数循环输出查询结果集中的数据。最后获取表单提交的数据，应用

mssql_field_name()函数获取指定字段的名称，其代码如下：

```
<?php
}
if (isset ( $_POST [sub] )) {
        if ($_POST [te] == "" || $_POST [te] == "输入数字") {
                echo "<script>alert('请在文本框中输入内容');</script>";
        } else {
                if (preg_match ( "/\d/", $_POST [te] )) {
                        if ($_POST [te] > 4) {
                                echo "<script>alert('输入的数字超出范围');</script>";
                        } else {
                                $rss = mssql_field_name ( $rs, $_POST [te] );
                        }
                } else {
                        echo "<script>alert('您输入的不是数字');</script>";
                }
        }
        ?>
<tr>
        <td bgcolor="#FFFFFF">数据表字段名称为</td>
        <td bgcolor="#FFFFFF" style="color: #FF0000"><?php echo $rss;?></td>
</tr>
<?php
        }
?>
```

秘笈心法

心法领悟 476：使用 mssql_field_name()函数的注意事项。

在使用此函数时，参数 field_index 从 0 开始。例如，第 3 个字段的索引值其实是 2，第 4 个字段的索引值是 3，依此类推。

实例 477 通过结果集对象输出数据

光盘位置：光盘\MR\10\477

高级

趣味指数：★★★★

实例说明

从结果集中获取数据的方法有很多，本实例主要讲解利用数据对象输出结果集中的数据。运行本实例，效果如图 10.10 所示。

通过结果集对象输出数据

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00

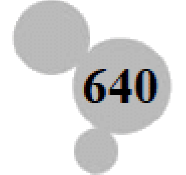
图 10.10　通过结果集对象输出数据

关键技术

mssql_fetch_object()函数从结果集中获取一行作为对象，如果没有更多行则返回 FALSE，其语法如下：

```
object mssql_fetch_object ( resource result )
```

该函数和 mssql_fetch_array()函数类似，只是返回一个对象而不是数组，即只能通过字段名来访问数据，而不是偏移量（数字是合法的属性名）。



设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）创建 index.php 文件，包含数据库连接文件，通过 mssql_query()函数执行查询操作，通过 while 语句和 mssql_fetch_object()函数循环输出查询结果集中的数据，其代码如下：

```
<?php
while ( $rst = mssql_fetch_object ( $rs ) ) {
?>
<tr>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk", "utf-8", $rst->id );  ?></td>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk", "utf-8", $rst->bname ); ?></td>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk", "utf-8", $rst->price );?></td>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk","utf-8",$rst->date);?></td>
</tr>
<?php
	}
?>
```

秘笈心法

心法领悟 477：使用 mssql_fetch_object()函数的注意事项。

使用 mssql_fetch_object()函数时需要注意，该函数返回的字段名是区分大小写的。

实例 478　通过结果集数组输出数据

光盘位置：光盘\MR\10\478

高级　趣味指数：★★★★

实例说明

实例 477 中讲解了通过结果集对象输出数据，下面将通过另外一个函数将结果集以数组的形式输出。运行本实例，结果如图 10.11 所示。

通过结果集数组输出数据

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00

图 10.11　通过结果集数组输出数据

关键技术

本实例主要应用 mssql_fetch_assoc()函数，从结果集中获取一行作为关联数组。如果没有更多行则返回 FALSE，其语法如下：

```
array mssql_fetch_assoc ( resource result_id )
```

参数 result_id 为指定的结果集。

设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）创建 index.php 文件，包含数据库连接文件，通过 mssql_query()函数执行查询操作，通过 while 语句和 mssql_fetch_assoc()函数循环输出查询结果集中的数据，其代码如下：

```
<?php
	include_once ("conn.php");
	$rs = mssql_query ( "select * from tb_demo10" );
?>
<table width="580px" bgcolor="#FC8C7E">
	<tr>
		<td style="color: #FFFFFF; text-align: center">图书 ID</td>
		<td style="color: #FFFFFF; text-align: center">图书名称</td>
		<td style="color: #FFFFFF; text-align: center">图书价格</td>
		<td style="color: #FFFFFF; text-align: center">出版日期</td>
	</tr>
<?php
	while ( $rst = mssql_fetch_assoc ( $rs ) ) {
?>
	<tr>
		<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [id] ); ?></td>
		<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [bname] ); ?></td>
		<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [price] );  ?></td>
		<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [date] ); ?></td>
	</tr>
<?php
	}
?>
```

秘笈心法

心法领悟 478：使用 mssql_fetch_assoc()函数的注意事项。

由于该函数返回的是一个关联数组，所以在获取数组中的元素时必须使用字段名称作为数组的索引，同时还要注意，其数组的索引是区分大小写的。

实例 479	返回结果集中的单元内容 光盘位置：光盘\MR\10\479	高级 趣味指数：★★★★

实例说明

如果用户只想查看用户名或者只想查看单一的一项数据，程序员就没有必要在与数据库的交互中取得一行数据，只需要从数据库中取得单一字段信息即可。本实例通过函数 mssql_result()实现获取结果数据，运行结果如图 10.12 所示。

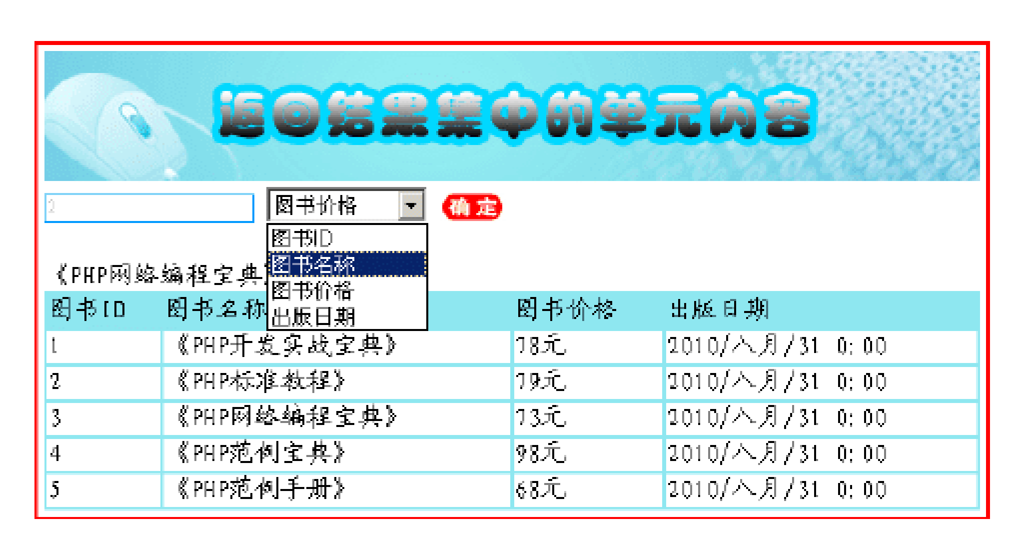


图 10.12　返回结果集中的单元内容

关键技术

本实例的关键点是如何运用 mssql_result()函数返回结果集中的单元内容，其语法如下：

```
string mssql_result ( resource result, int row, mixed field )
```

参数说明：

resource result：结果集。



关键技术

获取指定字段的长度应用的是 mssql_field_length()函数，其语法如下：

```
int mssql_field_length ( resource result, int field_offset )
```

参数说明：

resource result：结果集。

int field_offset：字段数。

设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）在 index.php 文件中创建 form 表单，提交指定要查询的字段值并且将数据提交到本页。

（3）在 index.php 文件中包含数据库连接文件，执行查询语句完成数据表中数据的循环输出，并对表单中提交的值进行判断，如果符合要求，则应用 mssql_field_length()函数获取表单提交的字段的长度值并且输出，其关键代码如下：

```
<?php
	include_once ("conn.php");
	$rs = mssql_query ( "select * from tb_demo10" );
	?>
		<table width="580px" bgcolor="#FEF1BA">
			<tr>
				<td style="color: #FFFFFF; text-align: center">图书 ID</td>
				<td style="color: #FFFFFF; text-align: center">图书名称</td>
				<td style="color: #FFFFFF; text-align: center">图书价格</td>
				<td style="color: #FFFFFF; text-align: center">出版日期</td>
			</tr>
	<?php
	if (isset ( $_POST [sub] )) {
		$rss = mssql_field_length ( $rs, $_POST [select] );
		echo "<font size='+1' color='red'>字段长度为" . $rss . "</font>";
	}
	while ( $rst = mssql_fetch_row ( $rs ) ) {
		?>
		<tr>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [0] ); ?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [1] ); ?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [2] ); ?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [3] ); ?></td>
		</tr>
<?php
	}
?>
```

秘笈心法

心法领悟 480：获取数据表中指定字段的类型。

在 mssql 函数库中，不但可以获取到数据表中指定字段的名称、长度、定位指针位置，而且可以获取到字段的类型，其应用的是 mssql_field_type()函数。

实例 481	利用函数返回指定字段的类型 光盘位置：光盘\MR\10\481	高级 趣味指数：★★★★

实例说明

通过上面的实例，相信读者已经掌握了如何利用函数获取字段的长度。下面将讲解如何利用函数获取指定



秘笈心法

心法领悟 481：SQL Server 数据库中的字段类型。

SQL Server 数据库中的数据类型与 MySQL 数据库中的数据类型有所不同，SQL Server 中的字段类型包括 int、real、string、blob、date 等。

实例 482	取得结果集中的字段数目 光盘位置：光盘\MR\10\482	高级 趣味指数：★★★★

实例说明

在 SQL Server 数据库中，取得结果集中的字段数目其实与操作 MySQL 数据库大同小异。下面就来讲述如何在 SQL Server 数据库中取得结果集中的字段数目，运行结果如图 10.15 所示。

取得结果集中的字段数目

获取字段数目

当前数据表字段数目为4

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00

图 10.15　获取结果集的字段数目

关键技术

获取结果集的字段数目主要应用 mssql_num_fields()函数，其语法如下：

```
int mssql_num_fields( resource result )
```

参数 resource result 指结果集。

设计过程

（1）新建 conn.php 文件，通过 mssql_connect()函数连接 SQL Server 数据库服务器，然后通过 mssql_select_db()函数连接 db_database10 数据库，其代码如下：

```
<?php
$conn = mssql_connect("localhost","sa","") or die ("连接 SQL Server 时发生错误");
mssql_select_db("db_database10",$conn) or die ("连接数据库发生错误");
?>
```

（2）创建 index.php 脚本文件，包含数据库连接文件，通过 mssql_query()函数执行查询语句，应用 mssql_num_fields()函数获取查询结果集中总的字段数，并且通过 while 语句和 mssql_fetch_row()函数完成查询结果集中数据的循环输出，代码如下：

```
<?php
	include_once ("conn.php");
	$rs = mssql_query ( "select * from tb_demo10" );
	if ($_GET [id] == 1) {
		$rss = mssql_num_fields ( $rs );
		echo "<font size='+1' color='red'>当前数据表字段数目为" . $rss . "</font>";
	}
	?>
		<table width="580px" bgcolor="#CCE587">
				<tr>
```



```
                                        <td style="color: #FFFFFF; text-align: center">图书 ID</td>
                                        <td style="color: #FFFFFF; text-align: center">图书名称</td>
                                        <td style="color: #FFFFFF; text-align: center">图书价格</td>
                                        <td style="color: #FFFFFF; text-align: center">出版日期</td>
                                </tr>
<?php
while ( $rst = mssql_fetch_row ( $rs ) ) {
        ?>
        <tr>
                                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [0] ); ?></td>
                                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [1] ); ?></td>
                                        <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [2] ); ?></td>
                                        <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [3] ); ?></td>
                                </tr>
<?php
        }
?>
```

秘笈心法

心法领悟 482：扩展知识。

与 mssql_num_fields()函数格式类似的 mssql_num_rows()函数，可用于获取数据表中结果集的记录数。

实例 483　获取数据表中数据记录数

光盘位置：光盘\MR\10\483

高级

趣味指数：★★★★

实例说明

在实例 482 的秘笈心法中笔者已经揭晓，获取数据表中数据记录数应用的是 mssql_num_rows()函数。下面将通过实例向读者讲解此函数的具体应用。运行本实例，效果如图 10.16 所示。

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00
数据表数据	信息数量为	5	

图 10.16　获取数据表中数据的记录数

关键技术

本实例主要应用 mssql_num_rows()函数实现获取数据表中数据的记录数，该函数的语法如下：

```
int mssql_num_rows ( resource result )
```

参数 result 为查询结果集。

设计过程

创建 index.php 脚本文件，包含数据库连接类，完成与 db_database10 数据库的连接，循环输出数据表中的数据，当单击“获取总数量”按钮时，输出数据表中总的记录数，代码如下：

```
<?php
        include_once("conn.php");                                                   //包含数据库连接文件
        $rs = mssql_query("select * from tb_demo10");
?>
```

设计过程

创建 index.php 脚本文件，包含数据库连接类，对类进行实例化操作，完成与 db_database10 数据库的连接并循环输出数据表中的数据，当单击“释放结果内存”超链接时，执行释放结果集的操作，代码如下：

```
<?php
	include_once("conn.php");                                    //包含数据库文件
	$ms = new Mssql("localhost","sa","","db_database10");        //实例化对象
	$rs = mssql_query("select * from tb_demo10");
?>
<?php
	while($rst = mssql_fetch_row($rs)){                           //循环输出
?>
<tr>
	<td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[0]);?></td>
	td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[1]);?></td>
	<td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[2]);?></td>
	<td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[3]);?></td>
</tr>
<?php
	}
	if($_GET[id] == 1){
		$ms -> free_result($rs);                                   //释放结果集
	}
?>
```

秘笈心法

心法领悟 484：详解 mssql_free_result()函数。

mssql_free_result()函数将释放所有与结果标识符 result 相关联的内存，仅需要在考虑到返回很大的结果集时会占用较多内存时调用，在脚本结束后所有关联的内存都会被自动释放。

实例 485　**mssql_close()函数的应用**　高级

光盘位置：光盘\MR\10\485　趣味指数：★★★★

实例说明

在程序运行结束后，不但要释放结果内存，还要关闭与数据库的连接以节省系统资源。运行本实例，单击“关闭连接”超链接将输出提示，运行结果如图 10.18 所示。

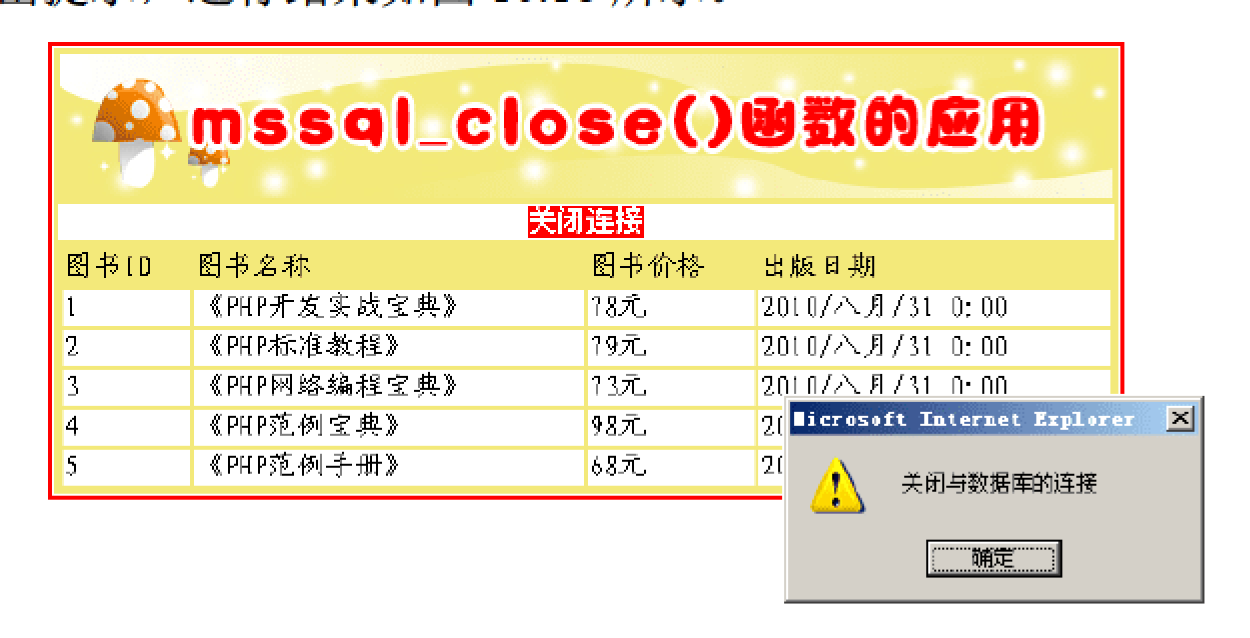


图 10.18　关闭与数据库的连接

关键技术

本实例主要是通过编写数据库类并通过对象句柄调用 mssql_close()函数，实现关闭数据库连接的操作。其



Server 数据库的连接，其中通过两种方法连接数据库，即通过 mssql_connect()函数实现与数据库的暂时连接、通过 mssql_pconnect()函数实现与数据库的永久连接。Mssql 类的代码如下：

```
<?php
class Mssql{
	var $host;
	var $user;
	var $pwd;
	var $conn;
	public function __construct($host,$user,$pwd,$conn){				//构造函数
		$this -> host = $host;
		$this -> user = $user;
		$this -> pwd = $pwd;
		$this -> conn = $conn;
		$this -> connect();
	}
	public function connect(){
		if($this -> conn == "pconn"){
			$this -> conn = mssql_pconnect($this -> host,$this -> user,$this -> pwd);
			echo "<script>alert('与 SQL Server 数据库建立永久连接');</script>";
		}else{
			$this -> conn = mssql_connect($this -> host,$this -> user,$this -> pwd);
			echo "<script>alert('与 SQL Server 数据库建立暂时连接');</script>";
		}
	}
}
?>
```

设计过程

（1）新建 conn.php 文件，编写数据库连接类，其完整代码请参考关键技术中的内容。

（2）新建 index.php 文件，创建 form 表单，提交连接数据库所使用的方式。然后包含数据库连接类，执行类的实例化操作，并调用 connect()方法完成与数据库的连接，其关键代码如下：

```
<?php
	include_once("conn.php");						//包含数据库连接类
	if(isset($_POST[sub])){						//判断按钮是否被单击
		if($_POST[select] == ""){					//判断下拉列表框内容
			echo "<script>alert('请选择连接方式');</script>";
		}else{
			$ms = new Mssql("localhost","sa","",$_POST[select]);	//实例化对象
		}
	}
?>
```

秘笈心法

心法领悟 486：对两种连接方式的选择。

永久连接和暂时连接从功能上说都可以实现与数据库的连接操作。至于如何选择，一般都是程序员的个人爱好问题。笔者喜欢用暂时连接的方式，虽然在操作完数据库后须及时关闭与数据库的连接，但是在结构上较为鲜明。

实例 487	添加图书类的实现 光盘位置：光盘\MR\10\487	高级 趣味指数：★★★★

实例说明

在实际 Web 开发过程中，无论是前台还是后台经常会用到向数据库中添加信息这一模块，如果所用之处都要重复编写代码，这会使整个项目的代码重用率大大降低。为了克服上述缺陷，可以通过一个数据信息添加

类来实现某项目中的所有信息的添加工作。运行本实例，图书添加页面及查看图书信息页面分别如图 10.20 和图 10.21 所示，首先在图 10.20 所示的表单中输入某图书的信息，单击“添加”按钮即可将图书信息添加到数据库中，同时可以在图 10.21 所示的页面中查看所有已经添加的图书信息。



图 10.20　添加图书信息



图 10.21　浏览图书信息

关键技术

本实例的关键是如何编写通用的添加信息类，其实现的基本原理是：通过构造函数对要执行的 SQL 语句进行初始化，这样即使表中的字段数目不同也可以实现将图书信息添加到数据库中，addbook 类的代码如下：

```
<?php
class addbook {
private $add_sql;                       //定义私有变量
public function __construct($x) {       //定义构造函数
        $this->add_sql = $x;            //为变量赋值
}
public function add() {                 //定义添加数据的方法
        include_once ("conn.php");
        if (@mssql_query ( $this->add_sql, $conn )) {
                echo "<script>alert('新书添加成功!');history.back();</script>";
        } else {
                echo "<script>alert('新书添加失败!');history.back();</script>";
        }

}
}
$bookname = $_POST [bookname];
$auto = $_POST [auto];
$pub = $_POST [pub];
$pubtime = $_POST [pubtime];
$sql = "insert into tb_demo08(bookname,auto,pub,pubtime)values('$bookname','$auto','$pub','$pubtime')";
$adbook = new addbook ( $sql );
$adbook->add ();
?>
```

该类定义一个私有的数据成员$add_sql 用于保存传入该类的查询语句，并通过构造函数__construct()实现对该私有成员的初始化，最后定义该类的成员函数 add()执行添加操作。

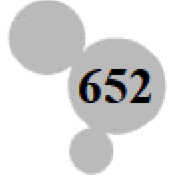
设计过程

（1）新建 conn.php 文件，完成与数据库的连接。

（2）新建 addbook.php 文件，创建 form 表单提交要添加的图书信息，将图书信息提交到 savebook.php 文件中。添加图书信息的页面效果如图 10.20 所示。

（3）创建 savebook.php 文件，编写数据添加类，定义 addbook()方法执行数据的添加操作。在该文件中完成类的实例化操作，并通过返回的对象调用类中的 addbook()方法，根据表单中提交的数据定义 insert 添加语句，最终执行数据的添加操作，其代码请参考关键技术中的内容。

（4）新建 index.php 文件，循环输出数据表中存储的图书信息，其关键代码如下：



```
        public function Delete(){                                   //定义删除方法
                if(!$rs = @mssql_query($this -> sql)){              //如果执行删除操作发生错误
                        echo "<script>alert('SQL 语句错误');</script>";     //JavaScript 提示
                }else{
                        echo "<script>alert('删除成功');location.href='index.php?id=1';</script>";
                }
        }
}
?>
```

设计过程

（1）新建 conn.php 文件，编写数据库连接类，完成与 SQL Server 数据库的连接，其代码如下：

```
<?php
class ConnDemo{                                                     //定义类名称
        var $host;                                                  //主机名称
        var $user;                                                  //用户名称
        var $pwd;                                                   //用户密码
        var $db;                                                    //数据库名称
        var $conn;                                                  //连接标识符
        var $tb;                                                    //数据表名称
        public function __construct($host,$user,$pwd,$db,$tb){      //构造函数
                $this -> host = $host;
                $this -> user = $user;
                $this -> pwd = $pwd;
                $this -> db = $db;
                $this -> tb = $tb;
                $this -> connect();
        }
        public function connect(){
                 $this -> conn = $conn;
                $this -> conn = mssql_connect($this -> host,$this -> user,$this -> pwd) or die ("Connect mssql false");
                mssql_select_db($this -> db,$this -> conn) or die ("Connect Database false");
        }
}
?>
```

（2）新建 delete.php 文件，编写数据库删除类并继承数据库连接类，其代码如关键技术中所示。

（3）新建 index.php 文件。首先创建 form 表单，提交图书的 ID 值作为执行删除操作的条件。然后包含数据删除类文件，执行 DeleteDemo 类的实例化操作并且向类中传递参数。接着执行查询语句，循环输出数据表中的数据。最后根据 form 表单提交的值，通过类实例化返回的对象调用 Delete()方法执行删除操作，其关键代码如下：

```
<?php
include ("delete.php");
$dd = new DeleteDemo ( "localhost", "sa", "", "db_database10", "tb_demo09" );
if ($_GET [id] == "1") {
 $rst = mssql_query ( "select * from tb_demo09" );
 ?>
<form action="" method="post">
<input type="text" size="10" name="text" value="输入 ID 删除" onfocus="this.value=''" class="one" />
        <input type="submit" name="sub" value=" " class="two" />
</form>
<table width="580px" bgcolor="#FA7672">
        <tr>
                <td align="center">图书名称</td>
                <td align="center">作者</td>
                <td align="center">出版社</td>
                <td align="center">出版时间</td>
        </tr>
<?php
 while ( $rstt = mssql_fetch_row ( $rst ) ) {
        ?>
        <tr>
                <td bgcolor="#FFFFFF" align="center"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [0] );?></td>
                <td align="center" bgcolor="#FFFFFF"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [1] );?></td>
                <td align="center" bgcolor="#FFFFFF"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [2] );?></td>
```

```
            <td align="center" bgcolor="#FFFFFF"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [3] );?></td>
        </tr>
<?php
}
}
if ($_POST [sub]) {                                              //判断提交按钮的值
        $te = $_POST[text];                                      //获取提交的 ID
        $dd -> SQL($te);                                         //调用 SQL 方法定义 delete 删除语句
        $dd -> Delete();                                         //调用 Delete()方法执行删除操作
}
?>
</td>
        </tr>
</table>
```

秘笈心法

心法领悟 488：类的继承。

少数面向对象的语言（如 C++、Smalltalk 等）支持多继承，即一个子类有多个父类。但是 PHP 自身不支持多继承，只支持单继承，即一个子类只有一个父类。

实例 489	更新图书类的实现 光盘位置：光盘\MR\10\489	高级 趣味指数：★★★★

实例说明

实现更新图书信息类与删除图书信息类大同小异，只是将删除图书信息类中的两个核心方法替换为更新操作的方法即可。运行本实例，执行图书信息的更新操作，其运行效果如图 10.23 所示。

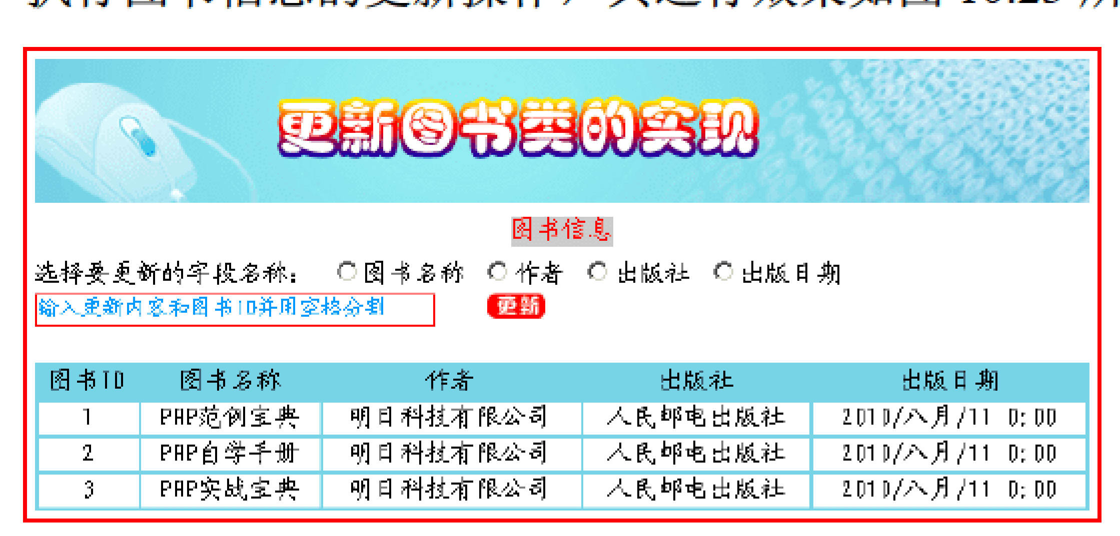


图 10.23　更新图书类的实现

关键技术

更新图书类 UpdateDemo 同样集成数据库连接类 ConnDemo，与数据库删除类的唯一不同之处是在 SQL 方法中定义的是 update 更新语句。UpdateDemo 类的代码如下：

```
class UpdateDemo extends ConnDemo{                                                    //继承抽象类
        var $sql;                                                                     //定义变量
        public function SQL($Condition1,$Condition2,$id){                             //定义方法拼接字符串
                $this -> sql = $sql;
                $this -> sql = "update ".$this -> tb." set $Condition1='$Condition2' where id=$id";    //拼接字符串
                return $this -> sql;
        }
        public function update(){                                                     //更新操作方法
                if($result = mssql_query($this -> sql)){
                        echo "<script>alert('更新操作以成功');location.href='index.php?id=1';</script>";    //JavaScript 提示
                }else{
                        echo "<script>alert('SQL 语句发生错误');</script>";
                }
```

秘笈心法

心法领悟 489：抽象类。

抽象类与普通的类在形式上没有太多的差别。抽象类中必须存在至少一个抽象方法，其好处在于被其他的子类继承时可以选择性地执行其中的一个或多个抽象方法。

实例 490　通过 ADO 方式实现输入页码跳转到指定页

光盘位置：光盘\MR\10\490

高级

趣味指数：★★★★

实例说明

分页显示模块的页码跳转方式有多种，比如单击页码跳转、从下拉列表框中选择页码实现跳转等。或许多数程序员已经熟悉通过函数方式实现分页，而本例将介绍如何通过 ADO 方式操纵 SQL Server 数据库实现分页显示。熟悉 ASP 的读者可能会发现，采用这种方式实现分页与 ASP 通过记录集实现分页的方式类似。运行本实例，如图 10.24 所示，首先在图中的文本框中输入要跳转的页码，然后单击 GO 按钮，即可实现页码跳转。

ADO方式实现指定页跳转

主题	发表时间	发表人	浏览次数
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56

共有主题406个 每页显示5个 第1页/共82页　1 2 3 4　GO

图 10.24　输入页码跳转到指定页

关键技术

为了方便操作，本实例将采用 GET 方法提交用于输入页码的表单，这样做的目的是使通过单击页码实现的跳转与通过在文本框中输入页码实现的跳转都用$_GET[]进行接收，否则还得单独做一个用$_POSI[]数组接收提交的页码的模块。通过 form 表单完成页码跳转的代码如下：

```
<form action="index.php" method="get" name="form1" id="form1">
<tr>
        <td width="30">
              <div align="center"><input type="text" name="page" size="2" class="inputcss" /></div>
        </td>
        <td width="40">
                <div align="center"><input name="submit" type="submit" class="buttoncss" value="GO" /></div>
        </td>
</tr>
</form>
```

设计过程

（1）新建 conn.php 文件，通过 ADO 连接 SQL Server 数据库，其代码如下：

```
<?php
$conn=new com("adodb.connection");                                                      //创建记录集
$connstr="provider=sqloledb;data source=localhost;uid=sa;pwd=;database=db_database10";   //连接 SQL Server 数据库
$conn->open($connstr);                                                                  //返回连接对象
?>
```

（2）编写代码用于接收查询字符串的值（该值代表要跳转的页）并实现分页显示，代码如下：

```
                                        for($i = 1; $i <= $rs->pagecount; $i ++) {
                                               ?>
        <a href="index.php?page=<?php  echo $i;?>"><?php  echo $i;?></a>
        <?php
                                        }
                                 } else {
                                        for($i = 1; $i <= 4; $i ++) {
                                               ?>
        <a href="index.php?page=<?php  echo $i;?>"><?php  echo $i;?></a>
        <?php
                                        }
                                        ?>
        <a href="index.php?page=<?php  if ($rs->pagecount >= $page + 1) echo $page + 1; else echo 1; ?>" title="后一页"><font face="webdings"> 8
</font></a>
        <a href="index.php?page=<?php  echo $rs->pagecount; ?>" title="尾页"><font face="webdings"> : </font></a>
        <?php
                                 }
                                 ?>
      </div>
        </td>
</tr>
</table>
```

秘笈心法

心法领悟 490：表单提交的两种方式。

提交表单可以采取 POST 和 GET 两种方式，这两种方式的区别是：使用 POST 方法提交的信息在地址栏中不显示，并且用$_POST[]数组接收提交的内容。而采用 GET 方法提交的信息在地址栏中显示，如果提交多个信息，则信息之间用“&”连接，并且提交的内容用$_GET[]数组进行接收。

实例 491　利用 mssql 函数库实现指定页跳转

光盘位置：光盘\MR\10\491

高级

趣味指数：★★★★

实例说明

通过单击页码跳转到指定页，是众多分页方式中应用最广泛的一种，通过这种方式可以浏览某一模块中的所有信息。PHP 操作 MySQL 数据库实现分页显示主要通过关键字 limit 实现，但 limit 关键字只是 MySQL 数据库的扩展部分，不能被 SQL Server 数据库识别，所以本实例将采取一种特殊方式实现 SQL Server 数据库信息的分页展示。运行本实例，如图 10.25 所示，通过单击图中页码超链接即可实现不同页之间的跳转。

指定页跳转

主题	发表时间	发表人	浏览次数
Java类相关图书	2007/十二月/6 0:00	who	56
明日科技出版的VB类相关图书	2007/十二月/5 0:00	lll	90
明日科技出版的C#类相关图书	2007/十二月/4 0:00	phk	12
明日科技出版的PB类相关图书	2007/十二月/3 0:00	zts	7
明日科技出版的C++类相关图书	2007/十二月/3 0:00	lzh	4

共有主题14个 每页显示5个 第1页/共3页　　1 2 3

图 10.25　利用 mssql 函数库实现指定页跳转

关键技术

本实例采用 mssql 函数库，通过面向过程的方式实现数据的分页输出，其基本原理是：在获取数据表中总的记录数后，通过标记变量$i 来控制显示记录的范围从而实现分页操作，其关键代码如下：

```
<?php
include_once ("conn.php");                                                    //包含数据库连接文件
$sql = mssql_query ( "select * from tb_demo07 order by pubtime desc", $conn );  //执行查询
$sql1 = mssql_query ( "select count(*) as total from tb_demo07", $conn );      //统计记录数
$info1 = mssql_fetch_array ( $sql1 );                                         //获取查询结果集
$total = $info1 [total];
if ($_GET [page] == "" || is_numeric ( $_GET [page] ) == false) {
        $page = 1;                                                            //定义分页变量
} else {
        $page = $_GET [page];                                                 //获取当前页码
}
$pagesize = 5;                                                                //定义每页显示记录数
if ($total % $pagesize == 0) {
        $totalpage = $total / $pagesize;
} else {
        $totalpage = ceil ( $total / $pagesize );
}
$i = 1;
while ( $info = mssql_fetch_array ( $sql ) ) {                                //循环输出查询结果
        if ($i > $pagesize * ($page - 1) && $i <= $pagesize * $page) {
        ?>
  <tr>
        <td height="25" bgcolor="#FFFFFF">
                <div align="left"><?php     echo iconv ( 'gbk', 'utf-8', $info [title] );?></div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
                <div align="center"><?php echo iconv ( 'gbk', 'utf-8', $info [pubtime] );?></div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
                <div align="center"><?php echo iconv ( 'gbk', 'utf-8', $info [pubuser] );?></div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
                <div align="center"><?php echo iconv ( 'gbk', 'utf-8', $info [click] );  ?></div>
        </td>
</tr>
  <?php
}
$i ++;
}
?>
```

设计过程

（1）新建 conn.php 文件，通过 mssql 函数库完成与 SQL Server 数据库的连接。

（2）创建 index.php 文件，包含数据库连接文件，查询指定数据表中的数据，并且统计总的记录数，定义分页变量，通过标记变量$i 来控制显示记录的范围，其代码请参考关键技术中的内容。

（3）创建分页超链接，完成数据的分页显示，其代码如下：

```
<table width="754" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
        <td width="338" height="25">
        <div align="left">共有主题<?php echo $total;?>个 每页显示<?php   echo $pagesize;?>个 第<?php echo $page;?>页/共<?php
echo $totalpage;?>页</div>
        </td>
        <td width="346">
        <div align="right">
<?php
for($i = 1; $i <= $totalpage; $i ++) {
        ?>
  <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
  <?php
}
?>
</div>
        </td>
        <td width="70"> </td>
</tr>
</table>
```



利用这种方法实现数据库信息的分页输出并不是将焦点放在数据库中，而是无论当前显示的是哪一页都从数据库中提取出所有的记录。

秘笈心法

心法领悟 491：分页原理。

在本实例中，只显示$pagesize*($page-1) 与$i<=$pagesize*$page 之间的记录，通过接收不同的$page 值来判断当前应显示哪一页。

10.3　SQL Server 数据库的综合应用

本节以 SQL Server 数据库为后台支持，将整个网站的制作过程分成若干模块进行逐一讲解。

实例 492	首页设计 光盘位置：光盘\MR\10\492	高级 趣味指数：★★★★

实例说明

“明日编程词典——在线体验版”网站首页应用框架设计，模拟编程词典软件的页面布局，其首页的设计效果如图 10.26 所示，其中包括网站模块导航、功能导航条、内容树状导航、内容展示区和滚动导航条。

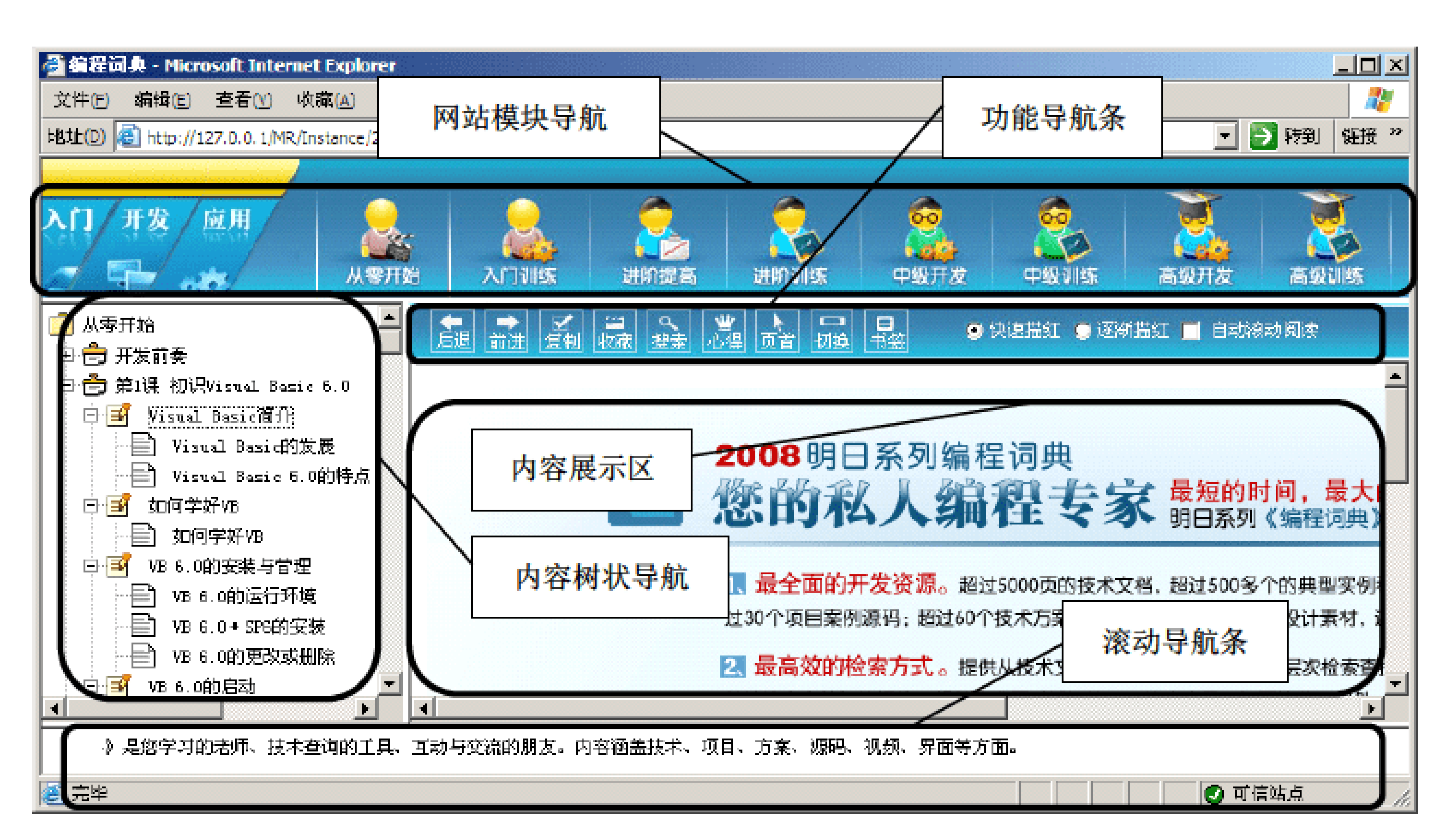


图 10.26　“明日编程词典——在线体验版”首页

关键技术

“明日编程词典——在线体验版”网站首页应用框架设计将文件分配为如下所示。

☑　框架主体文件：index.html、index.php。

☑　网站模块导航：pd_top.html、pd_top.php。

☑　内容树状导航：pd_left.html、pd_left.php。

☑　功能导航条：pd_title.html、pd_title.php。

☑　内容展示区：pd_main.html、pd_main.php。

☑　滚动导航条：pd_bottom.html、pd_bottom.php。

关键技术

网站模块导航中包含两项技术：

第一项技术：通过 js 脚本向 FLASH 中传递参数。

（1）创建 js 脚本文件 flash1.js，定义向 FLASH 中写入的内容，其代码如下：

```
function flash1(m){
 var flash2='<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,19,0" width="171" height="93">';
❶	flash2+='<param id="swf" name="movie" value="images/bccd.swf?conn_type='+m+'">';
	flash2+='<param name="quality" value="high">';
	flash2+='<param name="allScriptAccess" value="always"/>';
	flash2+='<param name="wmode" value="transparent">';
	flash2+='<embed src="images/bccd.swf" width="171" height="93" quality="high" pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash" wmode="transparent"></embed>';
	flash2+='</object>';
❷	document.write(flash2);
}
```



代码导读

❶ param id="swf"：在 swf 的 url 中传递参数值。

❷ document.write：执行写入操作。

（2）在 FLASH 元件中通过 on(release)方法获取传递的参数值，并执行超链接。

☑　通过 this.loaderinfo.parameters["conn_type"]获取传递的参数值，代码如下：

```
var type=this.loaderinfo.parameters["conn_type"];
```

FLASH 中的编辑效果如图 10.30 所示。

图 10.30　通过变量传递参数

☑　通过 getURL()方法执行超链接，代码如下：

```
getURL("pd_left.php?conn_id=20510111&table_name=study&class_type=从零开始&conn_type=type", "leftwindow", "GET");
```

FLASH 中的编辑效果如图 10.31 所示。

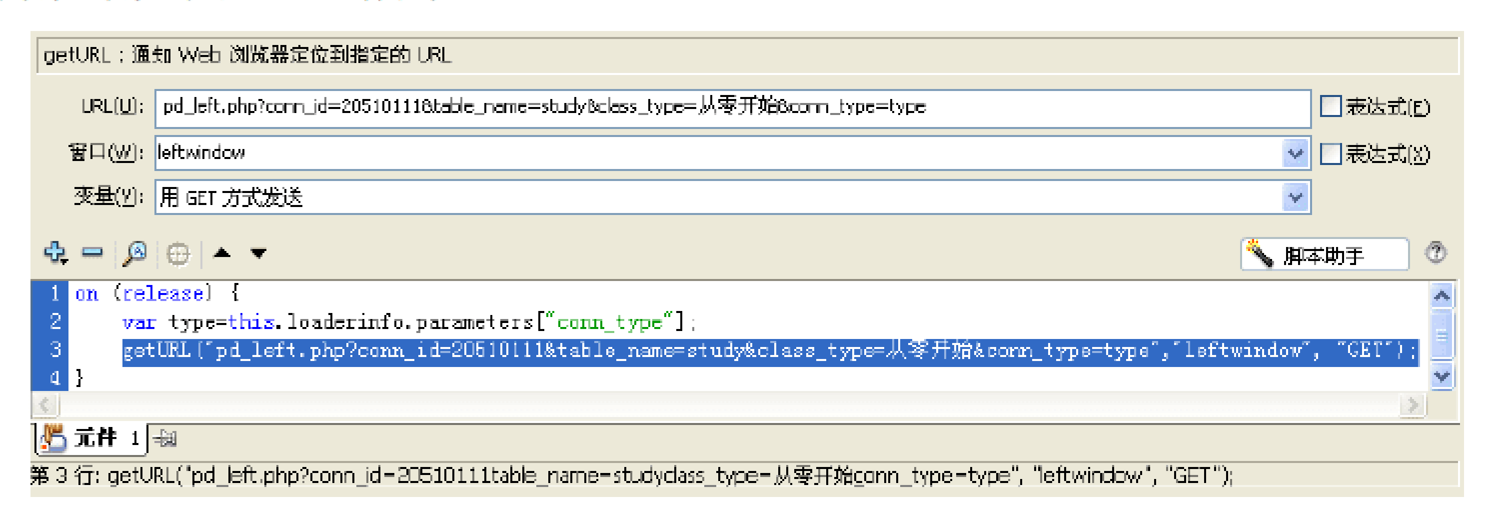


图 10.31　通过 getURL 执行超链接

（3）在本程序使用的 FLASH 中创建 3 个元件，分别对应入门、开发和应用 3 个模块。

元件 1 的脚本代码如下：

```
on (release) {
 var type=this.loaderinfo.parameters["conn_type"];
 getURL("pd_left.php?conn_id=20510111&table_name=study&class_type=从零开始&conn_type=type", "leftwindow", "GET");
}
```

元件 2 的脚本代码如下：

```
on (release) {
 var type=this.loaderinfo.parameters["conn_type"];
 getURL("pd_left.php?conn_id=10121002&table_name=jszx&class_type=技术中心&conn_type=type", "leftwindow", "GET");
}
```

元件 3 的脚本代码如下：

```
on (release) {
 var type=this.loaderinfo.parameters["conn_type"];
 getURL("pd_left.php?conn_id=10127105&table_name=ymzx&class_type=源码管理&conn_type=type", "leftwindow", "GET");
}
```

第二项技术：在超链接中应用 onMouseOver 和 onMouseOut 事件调用 JavaScript 脚本中的方法，控制鼠标滑过超链接图片时的样式。超链接的设置如下：

```
<a href="pd_left.php?conn_id=20510111&table_name=study&class_type=从零开始" target="leftwindow"><img
src="images/accidence/images_01.jpg" border="0" id="Image1" onMouseOver="MM_swapImage('Image1','','images/accidence/images_001.jpg',1)"
onMouseOut="MM_swapImgRestore()" /></a>
```

JavaScript 脚本中定义的方法如下：

```
function MM_preloadImages() { //v3.0
   var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
     var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
     if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
function MM_swapImgRestore() { //v3.0
   var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_findObj(n, d) { //v4.01
   var p,i,x;   if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
     d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
   if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
   for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
   if(!x && d.getElementById) x=d.getElementById(n); return x;
}
function MM_swapImage() { //v3.0
   var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
    if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
```

设计过程

网站模块导航的实现主要应用到 pd_top.html 和 pd_top.php，以及 js 文件夹下的 flash1.js 和 top.js 文件。

（1）创建 pd_top.php 文件，其主要功能是定义模板变量传递参数以及指定模板页，其代码如下：

```
<?php
session_start();                                         //初始化 SESSION 变量
require_once 'system/system.inc.php';                    //包含配置文件、数据库连接、操作文件
$smarty->assign('conn_id',$_GET['conn_id']);             //定义模板变量传递参数
$smarty->display('pd_top.html');                         //指定模板页
?>
```

（2）创建模板页 pd_top.html。首先通过 script 标签调用 js 脚本文件夹下的 top.js 和 flash1.js 文件。然后定义 JavaScript 脚本，执行 flash1()方法，向 FLASH 中传递参数值。最后根据模板变量$conn_id 的值进行判断，从而输出不同类别的功能模块，并为输出的功能模块创建超链接，链接到 pd_left.php 文件，同时传递相应的参数，其关键代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>{$conn_type}编程词典</title>
</head>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
❶<script type="text/javascript" src="js/top.js"></script>
❷<script src="js/flash1.js"></script>
❸<body onLoad="MM_preloadImages('images/criterion/images002.jpg')">
<div style="padding-top:inherit; width:100%; height: 93px; background:url(images/images_09.jpg);">
<div style="padding-top:inherit; width:171px; height: 93px; float:left; ">
<script type="text/javascript">
❹            flash1('{$conn_type}');
</script>
```

如果是应用类，内容树状导航中默认展示的是“源码管理”功能模块的内容，如图 10.34 所示。



图 10.33　开发类的内容树状导航



图 10.34　应用类的内容树状导航

关键技术

在内容树状导航中，根据网站模块导航中传递的参数值，控制框架中 topwindow（网站模块导航）、leftwindow（内容树状导航）和 mainwindow（内容展示区）3 部分内容的输出，其应用到的技术如下：

（1）switch 条件控制语句。

在 pd_left.php 文件中，通过$_GET 方法获取网站模块导航中参数 conn_id 传递的值，并将该值作为 switch 语句执行的条件，与 case 关键字指定的值进行判断，如果相同则执行相应的操作，首先将一个模板页赋给一个模板变量。然后通过 script 标签向 topwindow 中的 pd_top.php（网站模块导航）文件传递参数。最后通过 script 标签向 mainwindow 中的 pd_main.php（内容展示区）文件传递参数，其关键代码如下：

```
/*
switch 语句，判断 conn_id 的值是否与 case 关键字中指定的值相同，如果相同，则执行相应的操作
*/
switch($_GET[conn_id]){
case "20510111":                                        //指定关键字
        include 'accidence.php';                        //包含文件
        $smarty->assign('phtml','accidence.html');      //将模板文件的名称赋给模板变量
        //通过 script 脚本向指定的文件中传递参数
        echo "<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";
        echo "<script>parent.mainwindow.location.href='pd_main.php? conn_id=20510111';</script>";
        break;                                          //跳出循环
//省略了部分代码
```

（2）通过 Smarty 中的 include 函数在当前模板中包含其他模板文件。

在 switch 语句中，将指定的模板文件名称赋给一个模板变量 phtml，此时在内容树状导航的模板文件 pd_left.html 中，获取模板变量 phtml 传递的值，并且将该值作为 Smarty 中 include 函数包含的模板文件。内容树状导航模板文件 pd_left.html 的关键代码如下：

```
<script language="javascript" src="js/left.js"></script>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
<body>
<div style="width: 100%; height: 100%; padding-left:5px; padding-top:5px;">
{include file=$phtml}
</div>
</body>
```

（3）通过 Smarty 中的注册模板函数读取指定文件。

（4）通过 Smarty 中的 count_characters 和 truncate 方法统计和截取字符串。

在入门类、开发类和应用类的树状导航菜单中，都应用了 Smarty 中的 count_characters 和 truncate 方法，对指定字符串的长度进行统计和截取，从而判断数据属于几级菜单。count_characters 和 truncate 方法在本项目中应用的关键代码如下：

```
{if $subject[id12].8|count_characters=="12" && $subject[id12].8|truncate:8:""==$subject[id].8 }
```

上述代码取自 developer.html 模板文件。

设计过程

在内容树状导航概述中已经了解，内容树状导航除了包含 pd_left.php 和 pd_left.html 文件外，还包含入门、



开发和应用 3 类中的 6 个文件。

（1）在 pd_left.php 文件中，获取网站模块导航超链接中传递的参数值，并且将参数值赋给 SESSION 变量，通过 switch 语句对传递的参数值进行判断，从而控制框架中 topwindow（网站模块导航）、leftwindow（内容树状导航）和 mainwindow（内容展示区）3 部分内容的输出，其关键代码如下：

```
<?php
session_start();                                          //初始化 SESSION 变量
if($_GET[conn_id]=="" and $_GET[table_name]==""){         //判断 conn_id 和 table_name 的值为空
 $_SESSION[conn_id]="20510111";                           //设置初始值
 $_SESSION[table_name]="study";                           //设置数据库名称
 $_SESSION[class_type]="从零开始";                          //设置功能模块
}else{                                                    //如果不为空，则直接将它们的值赋给 SESSION 变量
 $_SESSION[conn_id]=$_GET[conn_id];
 $_SESSION[table_name]=$_GET[table_name];
 $_SESSION[class_type]=$_GET[class_type];
}
require 'system/system.inc.php';                          //包含数据库连接、操作方法和 Smarty 的配置方法
/*
switch 条件控制语句，判断 conn_id 的值是否与 case 关键字
中指定的值相同，如果相同，则执行相应的操作
*/
switch($_GET[conn_id]){
 case "20510111":                                         //指定关键字
        include 'accidence.php';                          //包含文件
        $smarty->assign('phtml','accidence.html');        //将模板文件的名称赋给模板变量
        //通过 script 脚本向指定的文件中传递参数
        echo "<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";
        echo "<script>parent.mainwindow.location.href='pd_main.php?conn_id= 20510111';</script>";
        break;                                            //跳出循环
 case "20610111":
        include 'accidence.php';
        $smarty->assign('phtml','accidence.html');
        echo "<script>parent.mainwindow.location.href='pd_main.php?conn_id= 26010111';</script>";
        break;
//省略了部分代码
 default:
        include 'accidence.php';
        $smarty->assign('phtml','accidence.html');
        echo "<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";                echo
"<script>try{parent.mainwindow.location.href='pd_main.php?conn_id= 20510111';}catch(e){}</script>";
        break;
}
$smarty->display('pd_left.html');                         //指定模板页
?>
```

（2）pd_left.html 模板文件中输出的内容由 switch 语句中定义的模板变量 phtml 的值决定，在 pd_left.html 文件中应用 Smarty 中的 include 函数包含由 phtml 指定的模板文件，其关键代码如下：

```
<script language="javascript" src="js/left.js"></script>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
<div style="width: 100%; height: 100%; padding-left:5px; padding-top:5px;">
{include file=$phtml}
</div>
```

（3）定义 developer.php 文件，获取 SESSION 变量中传递的值，执行查询操作，并将查询结果赋给指定的模板变量；注册模板函数，读取指定的文件，其代码如下：

```
<?php
if($_SESSION[conn_id]=="" and $_SESSION[table_name]==""){  //判断 SESSION 变量的值是否为空
 $conn_id="10127105";                                     //如果 SESSION 变量的值为空，则直接为变量赋值
 $table_name="ymzx";
 $class_type="源码管理";
}else{                                                    //如 SESSION 变量不为空，将 SESSION 的值赋给指定的变量
 $conn_id=$_SESSION[conn_id];
 $table_name=$_SESSION[table_name];
 $class_type=$_SESSION[class_type];
}
 $smarty->assign("conn_id",$conn_id);                     //设置模板变量
```

```
$smarty->assign("class_type",$class_type);                    //设置模板变量
function unHtml($params) {                                     //创建自定义函数
        extract($params);                                      //将数组中的数据转换到变量中
        if(strlen($text)==12){                                 //判断字符串的长度是否等于 12
                $str = is_file("files/".$text.".htm");         //判断指定的文件是否存在
                $str1 = is_file("files/".$text.".exe");        //判断指定的文件是否存在
                if($str){                                      //如果文件存在
                        return "files/".$text.".htm";          //返回文件名称
                }else if($str1){
                        return "files/".$text.".exe";          //返回文件名称
                }else{
                        return "files/1999.htm";               //否则返回一个新的文件
                }
        }else if(strlen($text)==16){                           //判断字符串的长度是否等于 16
                $str = is_file("files/".$text.".htm");         //判断指定的文件是否存在
                if($str){
                        return "files/".$text.".htm";          //返回文件名称
                }else if($str1){
                        return "files/".$text.".exe";          //返回文件名称
                }else{
                        return "files/2000.htm";               //否则返回一个新的文件
                }
        }
}
$smarty->register_function("Util", "unHtml");                  //注册模板函数
if($conn_id=="10121002@10122105@10123105@10125105"){           //判断变量的值
        $array_id=explode("@",$conn_id);                       //将 id 值返回到数组中
        $array_table=explode("@",$table_name);                 //将数据表名称的值返回到数组中
        for($i=0;$i<count($array_id);$i++){                    //循环读取数组中的 id 值
                //执行查询操作
                $rst[$i] = $admindb->ExecSQL("select * from $array_table[$i] where xxbh like '$array_id[$i]%'",$conn);
        }
        $result=array_merge($rst[0],$rst[1],$rst[2],$rst[3]);  //合并查询结果
        $smarty->assign('subject',$result);                    //将合并的查询结果赋给模板变量
}else{
        //执行查询操作
        $rst = $admindb->ExecSQL("select * from $table_name where xxbh like '$conn_id%'",$conn);
        $smarty->assign('subject',$rst);                       //将查询结果赋给模板变量
}
?>
```

（4）创建 developer.html 模板文件，通过 section 循环语句读取模板变量中存储的数据，并且应用 if 语句进行判断，如果指定字符串的长度是 8 个字节，则说明是一级导航菜单；如果指定字符串的长度是 12 个字节，并且其前 8 个字节与一级导航菜单的 8 个字节相同，那么它是二级导航菜单；如果指定字符串的长度是 16 个字节，并且其前 8 个字节与一级导航菜单的 8 个字节相同，其前 12 个字节与二级导航菜单的 12 个字节相同，那么它是三级导航菜单。具体代码请参考光盘中内容。

秘笈心法

心法领悟 494：使用 pd_left.html 模板文件的注意事项。

在 pd_left.html 模板文件中，必须载入 left.js 脚本文件和 link.css 样式文件，这是实现树状导航菜单功能和控制页面属性的关键。

实例 495　内容展示区设计

光盘位置：光盘\MR\10\495

高级　趣味指数：★★★★

实例说明

内容展示区可输出内容树状导航菜单中指定标题的完整内容。如果指定标题的完整内容不存在，则输出指



定的替代文件，其运行效果如图 10.35 所示。

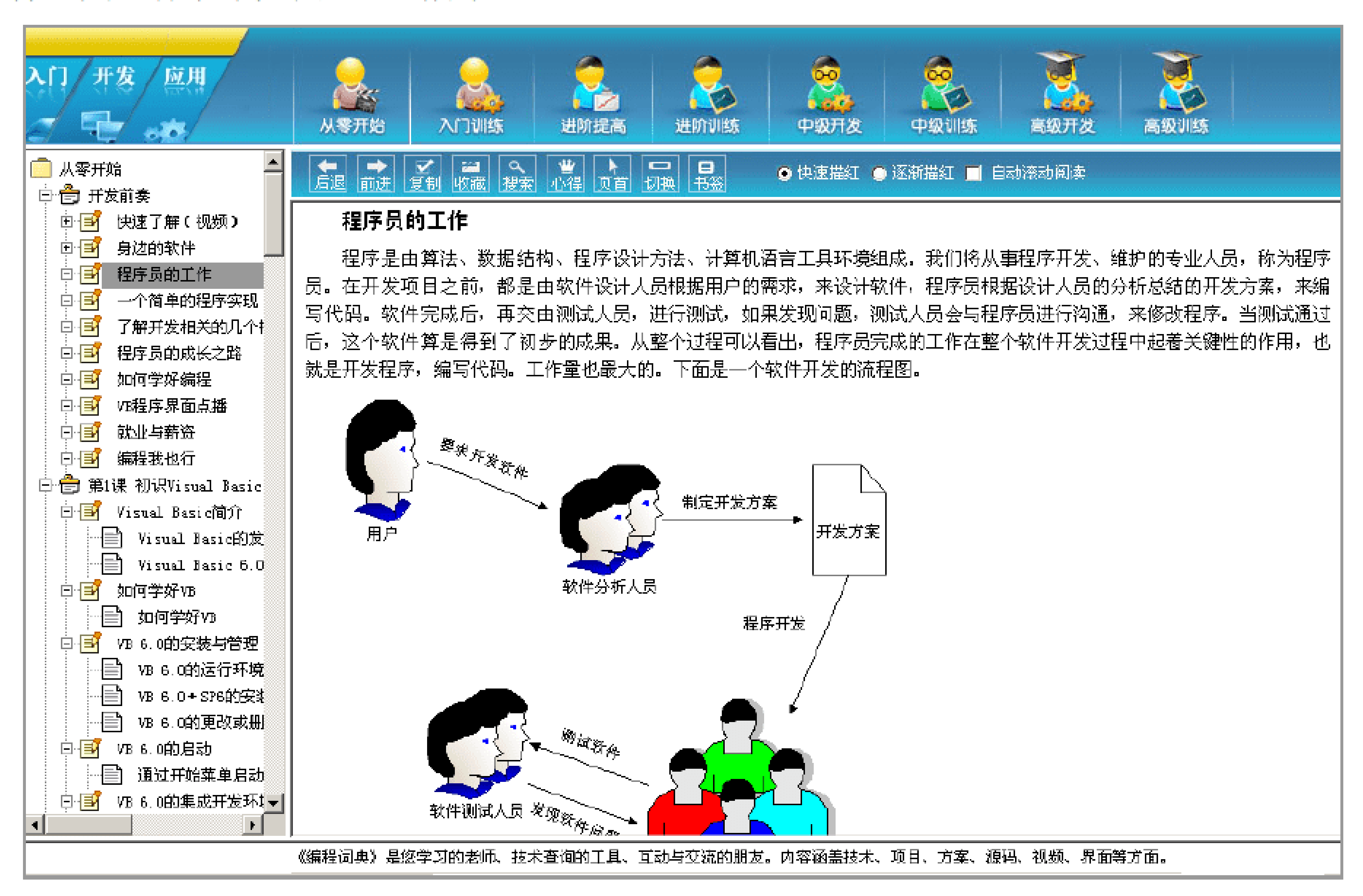


图 10.35　内容展示区

关键技术

内容展示区中完整内容的输出，关键是内容树状导航菜单中标题的超链接值的设置，该值是从注册模板函数中获取的。也就是说在注册模板函数中判断指定标题的完整文件是否存在，如果存在，则输出这个文件，否则输出指定的替代文件。在本项目中定义 3 个注册模板函数，其功能都是输出指定的文件，这 3 个模板文件分别存储于 accidence.php、criterion.php 和 developer.php 中。

设计过程

内容展示区只包含一个文件 pd_main.php。在该文件中判断超链接传递的参数值，并根据此数值包含不同的替代文件，其代码如下：

```
<?php
//require_once 'system/system.inc.php';                          //调用配置文件
//$smarty->assign('conn_id',$_GET[conn_id]);
if($_GET[conn_id]=="20510111" or $_GET[conn_id]=="10121605"){     //判断超链接的 id 值
 include_once("files/1999.htm");                                 //调用指定的包含文件
}else{
 include_once("files/2000.htm");                                 //调用指定的包含文件
}
?>
```

秘笈心法

心法领悟 495：关于 JavaScript。

在“明日编程词典——在线体验版”的页面设计中应用了很多 JavaScript 代码，其中包括对导航栏和树状导航栏的设计。如比较流行的 JQuery 框架即是业界追捧的 JavaScript 框架，如果用户有 JavaScript 脚本语言的基础，可以利用 JQuery 框架实现本实例的内容。

10.4 SQL Server 数据库的安全与维护

实例 496 压缩数据库 光盘位置：光盘\MR\10\496 高级 趣味指数：★★★★

实例说明

数据库在使用一段时间后，时常会出现因删除数据而造成的空闲空间太多的情况，这时就需要减少分配给数据库文件和事务日志文件的空间，以免浪费磁盘空间。当数据库中没有数据时，可以直接修改数据库文件的属性以改变其占用空间，但当数据库中有数据时，这样操作就会破坏其中的数据，因此需要使用收缩的方式来缩减数据库空间。本实例实现将 SQL Server 数据库 model 的未使用空间缩为数据库大小的 10%。在查询分析器中的运行结果如图 10.36 所示。

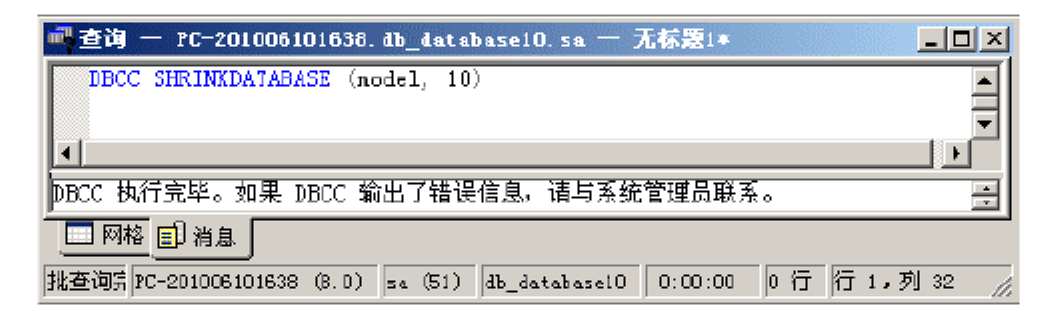


图 10.36 收缩数据和日志

关键技术

可以使用 DBCC SHRINKDATABASE 命令来压缩数据库。

DBCC SHRINKDATABASE 语句的语法如下：

```
DBCC SHRINKDATABASE
    ( database_name [ , target_percent ]
        [ , { NOTRUNCATE | TRUNCATEONLY } ]
    )
```

参数说明：

database_name：要压缩的数据库名称。

target_percent：数据库压缩后，数据库文件中所要的剩余可用空间百分比。如果指定的百分比过大，超过了压缩前未使用的空间所占的比例，则数据库不会被压缩。另外，压缩后的数据库不能比数据库初使设定的容量小。

NOTRUNCATE：在数据库文件中保留所释放的文件空间。如果未指定，将释放的文件空间分配给操作系统。

TRUNCATEONLY：将数据库文件中的任何未使用的空间释放给操作系统，并将文件收缩到上一次所分配的大小，从而减小文件而不移动任何数据，不试图重新定位未分配页的行。选择此项后，target_percent 选项将无效。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在查询分析器中选择要连接的数据库为 db_database10。

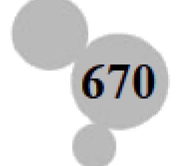
（3）在代码编辑区中输入 SQL 语句：

```
DBCC SHRINKDATABASE (model, 10)
```

（4）单击执行按钮 ▶，执行本实例，即可将数据库压缩。

秘笈心法

心法领悟 496：压缩数据库。



（3）单击执行按钮▶，执行本实例，即可实现数据的收缩。

秘笈心法

心法领悟 497：收缩数据与压缩数据。

在笔者看来，收缩数据与压缩数据在功能上没有什么不同。压缩数据是按比例将数据的物理内存减小，而收缩数据是根据传入参数将数据文件的大小减少至一定值。

实例 498	备份数据库 光盘位置：光盘\MR\10\498	高级 趣味指数：★★★★

实例说明

在网站开发过程中，数据库一般都保存有许多重要的数据信息，这些数据的丢失将会带来很严重的后果，所以对系统数据库进行备份非常重要，这样可以在数据库出现错误时能够及时地恢复数据库。本实例介绍使用 SQL 语句备份 SQL Server 数据库的过程，并使 SQL Server 在完成备份的 20%时显示备份进度信息。

关键技术

本实例使用 BACKUP DATABASE 语句，完成数据库、差异式或文件和文件组的备份，其简要用法如下：

```
BACKUP DATABASE { database_name | @database_name_var }
TO < backup_device > [ ,...n ]
[ WITH
[ BLOCKSIZE = { blocksize | @blocksize_variable } ]
     [ [ , ] NAME = { backup_set_name | @backup_set_name_var } ]
     [ [ , ] DESCRIPTION = { 'text' | @text_variable } ]
     [ [ , ] DIFFERENTIAL ]
     [ [ , ] FORMAT | NOFORMAT ]
     [ [ , ] { NOUNLOAD | UNLOAD } ]
     [ [ , ] RESTART ]
     [ [ , ] STATS [ = percentage ] ]
]
```

BACKUP DATABASE 语句参数说明如表 10.2 所示。

表 10.2　BACKUP DATABASE 语句的参数说明

参　　数	说　　明
BLOCKSIZE	设置写入时的数据区块大小。此值通常不需设置，可以由 SQL Server 自行决定，但如要将备份数据库刻录到光盘上，并打算用光盘进行还原，则需将此值设置为 2048。如要指定磁盘的数据区块大小，必须搭配 FORMAT 参数使用
NAME	表示备份结果集的名称，其长度不超过 128 个字符
DESCRIPTION	描述备份设置的文本，最多包括 255 个字符
DIFFERENTIAL	表示仅备份上次进行完整数据库备份以来数据所发生的变化，即进行差异备份。差异备份一般比完整备份占用更少的空间。对于上一次完整备份时备份的全部单个日志，使用该选项可以不必再进行备份
FORMAT|NOFORMAT	设置是否先格式化磁盘
NOUNLOAD|UNLOAD	设置是否要在备份完成时退出磁盘
RESTART	指定 SQL Server 重新启动一个被中断的备份操作。因为 RESTART 选项在备份操作中断处重新启动该操作，所以节省了时间。若要重新启动一个特定的备份操作，要重复整个 BACKUP 语句并且加入 RESTART 选项。不一定非要使用 RESTART 选项，但它具有节省时间的优势
STATS	表示在备份完某一百分比数据之后所要显示的信息

设计过程

下面将备份 db_database10 数据库，具体操作步骤如下：

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
BACKUP DATABASE db_sql
TO disk='E:\MyData\db_sql 备份.bak'
```

（3）单击执行按钮▶，执行备份。备份后的 db_database10 数据库被保存在 E 盘的 SQL Server Data 文件夹中，如图 10.38 所示。



图 10.38　备份数据库文件

秘笈心法

心法领悟 498：备份数据库的注意事项。

笔者在讲解 MySQL 数据库备份时已经说过，备份出来的文件虽然已经经过加密处理，但也是将数据信息以近乎明文的方式展现给用户，存在一定的风险。

实例 499　备份事务日志　高级

光盘位置：光盘\MR\10\499　趣味指数：★★★★

实例说明

本实例实现的是在创建的命名备份设备上创建 db_database10 数据库的事务日志备份。事务日志备份仅制作日志文件的复本。日志文件备份本身不能用于还原数据库。日志文件在数据库还原后，可将数据库恢复到原始的故障点。

关键技术

本实例应用的技术有创建备份设备和应用 BACKUP BLOG 语句备份事务日志。

1．创建备份设备

备份设备是创建备份和恢复数据库的前提条件，可以将数据库、事务日志和文件备份到磁盘和磁带上。设备可以分为 4 种：磁盘设备、磁带设备、物理设备和逻辑设备。

磁盘设备、磁带设备和物理设备在备份时，只要选择相应的设备即可。而逻辑设备用来标识物理设备的别名或公用名称，其名称永久地存储在 SQL Server 内的系统表中。使用逻辑备份的优点是引用名称简单，例如，逻辑设备名称可以是 mrsoft_Backupfdw，而物理设备名称则是 E:\SQL Server Data\mrosft\MyBackup.bak。

在 Transact-SQL 语句中，使用 sp_addumpdevice 系统存储过程创建备份设备，语法如下：

```
sp_addumpdevice [ @devtype = ] 'device_type' ,
      [ @logicalname = ] 'logical_name' ,
      [ @physicalname = ] 'physical_name'
      [ , { [ @cntrltype = ] controller_type
                | [ @devstatus = ] 'device_status'
            }
      ]
```

参数说明：

device_type：备份设备的类型。device_type 的数据类型为 varchar(20)，没有默认设置，可以是 Disk（硬盘文件）、Pipe（命名管理）或 Tape（磁带设备）。

logicalname：备份设备的逻辑名称，该逻辑名称用于 BACKUP 和 RESTORE 语句中，其数据类型为 sysname，没有默认值，并且不能为 NULL。

physicalname：备份设备的物理名称。物理名称必须遵照操作系统文件名称的规则或者网络设备的通用命名规则，并且必须包括完整的路径。physical_name 的数据类型为 nvarchar(260)，没有默认值并且不能为 NULL。

2. 用 BACKUP LOG 语句备份事务日志

执行 BACKUP LOG 语句备份事务日志，同时指定要备份的事务日志所属的数据库名称、事务日志备份将写入的备份设备。其语法格式请参见实例 498 中的技术分析。

执行完整数据库备份或差异数据备份期间不能备份事务日志，但在运行文件备份的同时可以备份事务日志。另外，下列情况不要备份事务日志：

（1）在由于事务日志包含自上次创建备份后数据库所发生的更改，而创建了数据库或文件备份之前。

（2）事务日志已被截断。此时如果要备份事务日志，必须先创建数据或差异数据库备份。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句，创建一个名为 mrsoft_Backupfdw 的逻辑磁盘设备，其物理路径为 E:\SQL Server Data\mrosft\MyBackup.bak。

```
SP_ADDUMPDEVICE 'disk','mrsoft_Backupfdw',' E:\SQL Server Data\mrosft\MyBackup.bak '
```

（3）单击执行按钮▶，执行本实例，其运行结果如图 10.39 所示。

（4）在代码编辑区中输入如下 SQL 语句，用来在创建的命名备份设备 mrsoft_Backupfdw 上创建 db_database10 数据库的事务日志备份。

```
BACKUP LOG db_database10
TO mrsoft_Backupfdw
GO
```

（5）单击执行按钮▶，执行本实例，其运行结果如图 10.40 所示。

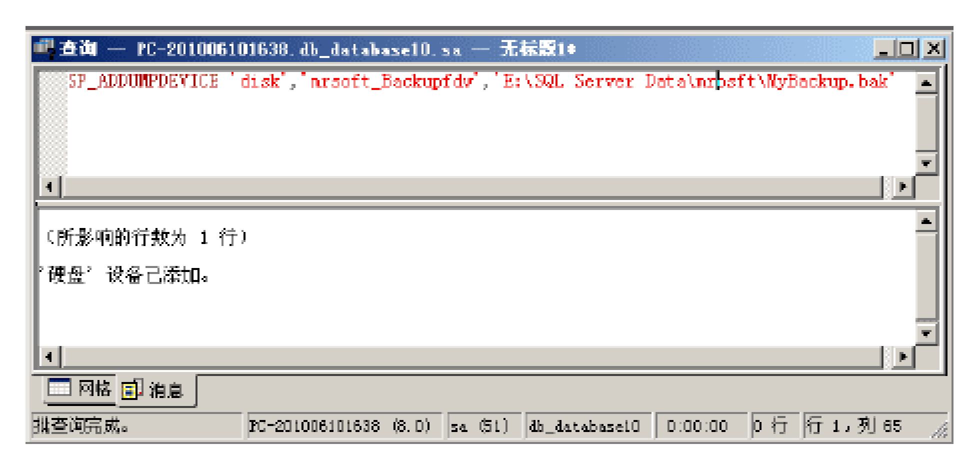


图 10.39　创建备份设备

BACKUP LOG 操作成功地处理了 0 页，花费了 0.027 秒（0.000 MB/秒）。

图 10.40　事务日志备份

秘笈心法

心法领悟 499：SQL Server 备份事务日志的注意事项。

在 SQL Server 中备份 db_database10 数据库事务日志的过程中，首先选择数据库的故障还原模式，如选择“完全”模式，然后单击“确定”按钮，如图 10.41 所示。

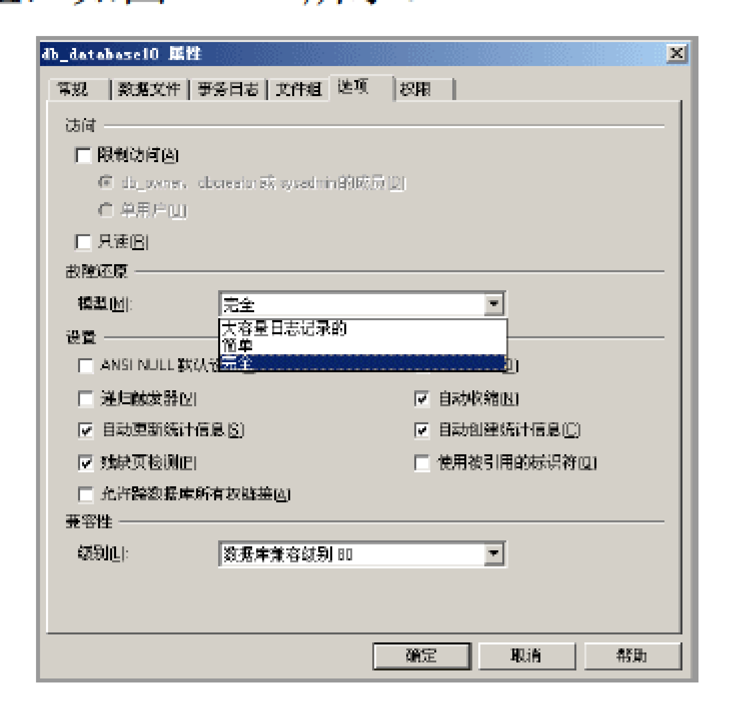
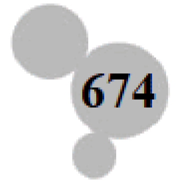


图 10.41　db_database10 数据库故障还原模式



实例 500　**检查备份集**　高级

光盘位置：光盘\MR\10\500　趣味指数：★★★★

实例说明

本实例首先备份一个名为 db_database10 的数据库，然后检查该数据库备份集是否完整以及所有卷是否都可读。在查询分析器中的运行结果如图 10.42 所示。

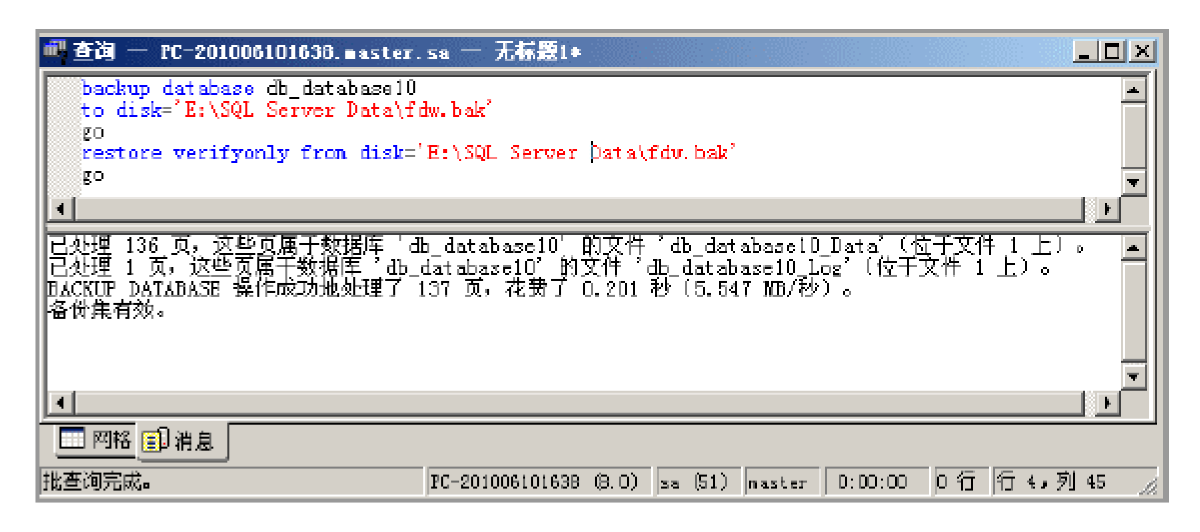


图 10.42　检查备份集

关键技术

本实例主要应用 RESTORE VERIFYONLY 语句检查数据库备份集是否有效，其语法如下：

```
RESTORE VERIFYONLY
FROM < backup  device > [ ,...n ]
[ WITH
      [ FILE = file_number ]
      [ [ , ] { NOUNLOAD | UNLOAD } ]
      [ [ , ] LOADHISTORY ]
      [ [ , ] PASSWORD = { password | @password_variable } ]
      [ [ , ] MEDIAPASSWORD = { mediapassword | @mediapassword_variable } ]
      [ [ , ] { NOREWIND | REWIND } ]
]
```

参数说明：

backup_device：指定还原时要使用的逻辑或物理备份设备。

FILE = file_number：标识要还原或处理的备份集。例如，file_number 为 1 表示第 1 个备份集，file_number 为 2 表示第 2 个备份集。如果没有指定 file_number，则假定是指定 <backup_device>上的第 1 个备份集。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
backup database db_database10
to disk='E:\SQL Server Data\fdw.bak'
go
restore verifyonly from disk='E:\SQL Server Data\fdw.bak'
go
```

（3）单击执行按钮▶，执行本实例，其运行结果如图 10.42 所示。

秘笈心法

心法领悟 500：检查备份集的相关事项。

RESTORE VERIFYONLY 语句用来验证备份但不还原备份，它检查备份集是否完整以及所有卷是否都可

用 style 参数转换日期的语句如下：

```
SELECT CONVERT(char(12),GETDATE(),3)
```

此语句把当前日期转换为样式 3，即 dd/mm/yy。

实例 502　数据库中数据的加密

光盘位置：光盘\MR\10\502

高级

趣味指数：★★★★

实例说明

本实例实现对 db_database10 数据库 tb_demo11 用户信息表中的用户名和密码进行加密。注意，在通过 pwdencrypt()函数对相同的用户名和密码进行加密后，生成的结果不相同并且是不可逆的。本实例的运行结果如图 10.45 所示。

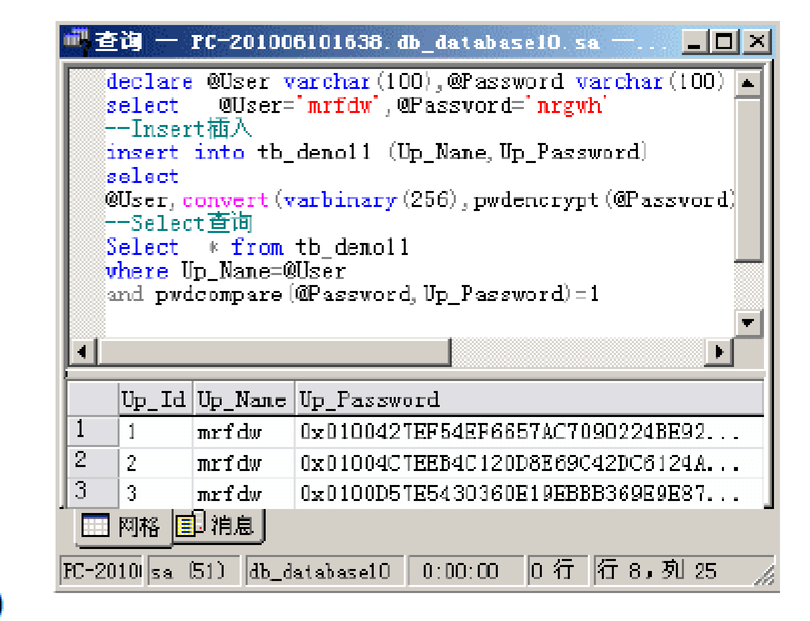


图 10.45　对数据库中的数据进行加密

关键技术

本实例主要应用加密函数 pwdencrypt()和加密校验函数 pwdcompare()实现。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
declare @User varchar(100),@Password varchar(100)
select     @User='mrfdw',@Password='mrgwh'
--Insert 插入
insert into dbo.tb_demo11 (Up_Name,Up_Password)
select
@User,convert(varbinary(256),pwdencrypt(@Password))
--Select 查询
Select   * from tb_demo11
where Up_Name=@User
and pwdcompare(@Password,Up_Password)=1
```

（3）单击执行按钮 ▶，执行本实例，其实现的结果如图 10.45 所示。

秘笈心法

心法领悟 502：加密函数 pwdencrypt()。

由于此函数对插入的相同数据进行加密后，生成的结果不相同而且不可逆，是一种非常安全的加密手段，被广泛应用。

实例 503　SQL Server 数据库的分离和附加

光盘位置：光盘\MR\10\503

高级

趣味指数：★★★★

实例说明

为了方便读者运行和学习本章实例内容，避免出现不必要的操作失误，特向读者介绍一些操作 SQL Server 数据库的细节内容。本实例主要讲解如何分离、附加数据库。

关键技术

本实例主要应用 xp_cmdshell 函数和 bcp 实用工具实现功能。

（1）xp_cmdshell 函数

xp_cmdshell：以操作系统命令行解释器的方式执行给定的命令字符串，并以文本行方式返回任何输出，其语法如下：

```
xp_cmdshell {'command_string'} [, no_output]
```

参数说明：

'command_string'：在操作系统命令行解释器上执行的命令字符串。command_string 的数据类型为 varchar(255) 或 nvarchar(4000)，没有默认值。command_string 不能包含一对以上的双引号，如果所引用的文件路径或程序名称中有空格，则需要使用一对引号。

no_output：可选参数，表示执行给定的 command_string，但不向客户端返回任何输出。

返回代码值：0（成功）或 1（失败）。

（2）bcp 实用工具

bcp 实用工具是一种命令提示符大容量复制的实用工具，它以用户指定的格式从操作系统文件或向操作系统文件复制。下面以 bcp 实用工具用于 SQL Server 2000 实例和数据文件之间指定的格式复制数据为例，介绍 bcp 实用工具中几个重要的参数。

☑ In|out|queryout|format：指定大容量复制的方向。In 是指从文件复制到数据库表或视图，out 是指从数据库表或视图复制到文件。
☑ -c：使用字符数据类型执行大容量复制操作。
☑ -q：在 bcp 实用工具和 SQL Server 实例的连接中执行 SET QUOTED_IDENTIFLERSON 语句。
☑ -S server_name[/instance_name]：指定要连接的 SQL Server 实例。
☑ -U login_id：指定用于连接的 SQL Server 实例。
☑ -P password：指定登录 ID 的密码。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
EXEC master..xp_cmdshell 'bcp db_database10.dbo.tb_demo10 out E:\tb_demo10.xls -c -q -S"localhost"
-U"sa" -P""'
```

（3）单击执行按钮，执行本实例，其运行结果如图 10.49 所示。

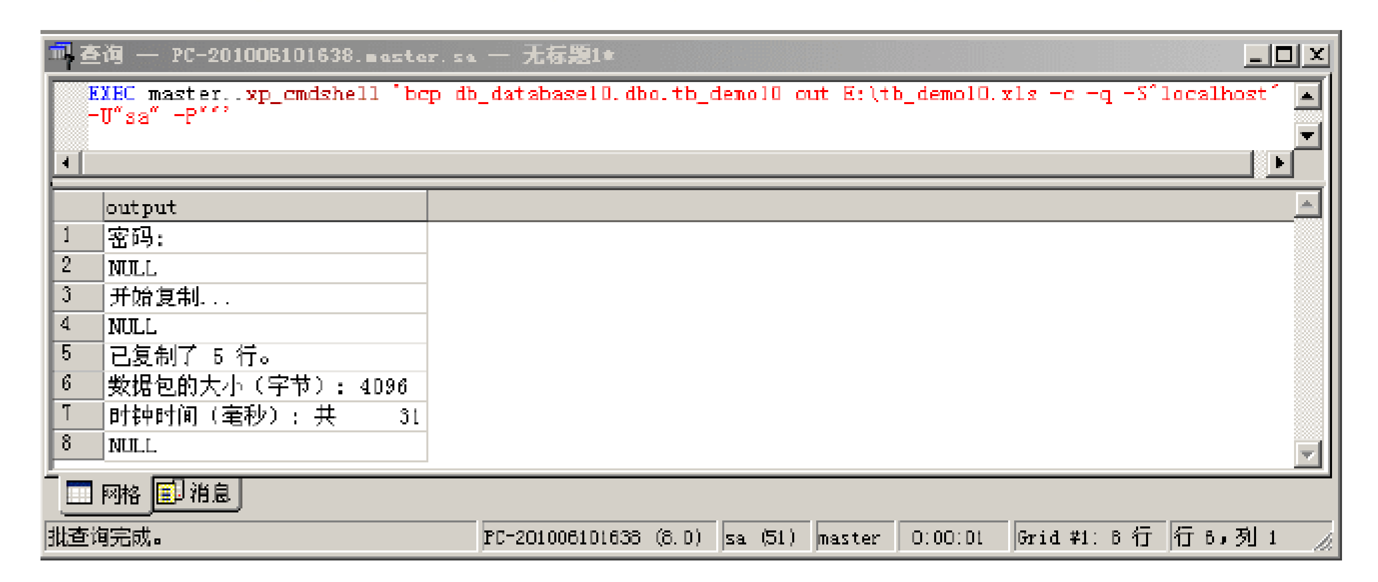


图 10.49　从 SQL Server 数据库导出数据至 Excel 表中

秘笈心法

心法领悟 504：使用 xp_cmdshell 函数的注意事项。



xp_cmdshell 以同步方式操作，在命令行解释器命令执行完毕之前，不会返回控制。

实例 505	将查询到的数据生成一个 Excel 文件 光盘位置：光盘\MR\10\505	高级 趣味指数：★★★★

实例说明

本实例讲解将 SQL Server 中 db_database10 数据库的 tb_demo10 数据表中的查询信息生成一个存储到本地 E 盘的 tb_mrsoft.xls 文件。

关键技术

本实例同样是应用 xp_cmdshell 函数和 bcp 实用工具实现，相关内容请读者参考实例 504。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在查询分析器中选择要连接的数据库为 db_database10。

（3）在代码编辑区中输入 SQL 语句：

```
declare @sql varchar(1000)
set @sql = 'bcp
"SELECT * FROM db_database10..tb_demo10"
queryout    E:\SQL Server Data\tb_mrsoft.xls -c -q -S"localhost"
-U"sa" -P""'
EXEC master..xp_cmdshell @sql
```

（4）单击执行按钮▶，执行本实例，其运行结果如图 10.50 所示。

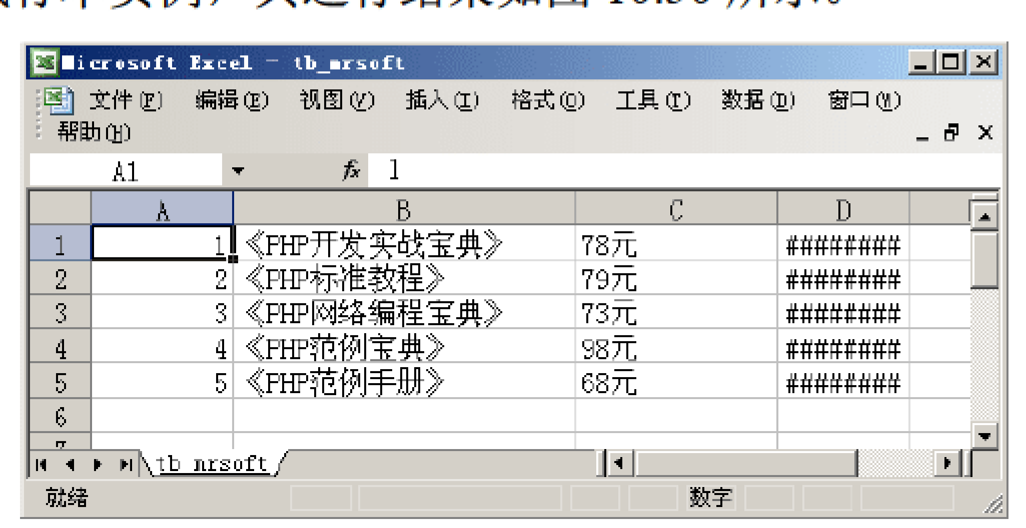
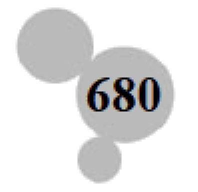


图 10.50　生成 Excel 文件

秘笈心法

心法领悟 505：将数据导入到 Excel 文件中。

将数据导入到 Excel 中是程序员常用的一种数据统计方法，因为 Excel 中对数据的求和、排序等有良好的支持，而且将数据库信息存储到 Excel 中也是一种备份数据库的方法，只是这种备份方法并不常用。



第 11 章

PHP 操作 Access 数据库

- 连接数据库
- 操作 Access 数据库
- Access 数据库的安全
- Access 数据库的综合运用

11.1　连接数据库

如果开发一个小型网站，利用 Access 数据库不仅方便操作、易于维护，更重要的是可以节省服务器资源，下面将通过具体实例讲解 PHP 操纵 Access 数据库的方法。

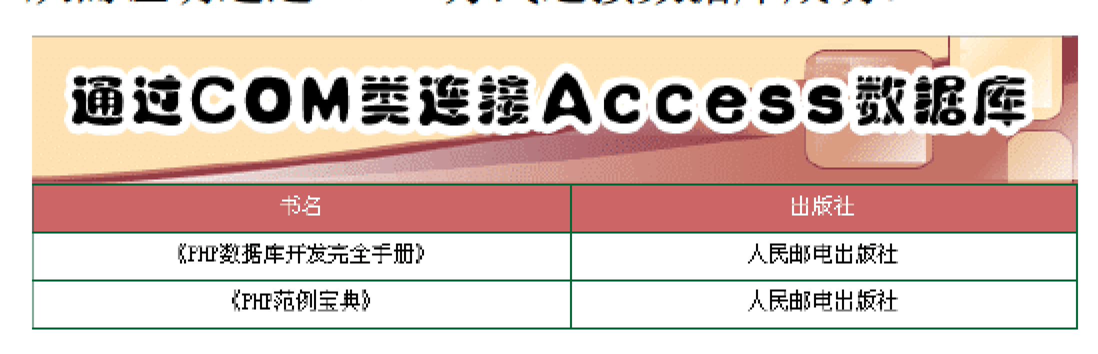
实例 506　通过 COM 类连接 Access 数据库

光盘位置：光盘\MR\11\506

高级　趣味指数：★★★★

实例说明

本实例主要介绍 PHP 如何通过 ADO 方式连接 Access 数据库，效果如图 11.1 所示，该页面显示的信息是从 Access 数据库中检索出来的，从而证明通过 ADO 方式连接数据库成功。



书名	出版社
《PHP数据库开发完全手册》	人民邮电出版社
《PHP范例宝典》	人民邮电出版社

图 11.1　连接 Access 数据库

关键技术

ADO 技术基于通用对象模型（COM），可提供多种语言的访问技术。PHP 是通过预先定义类 COM 来使用 ADO 方法操纵 Access 数据库的，该类的详细说明如下：

```
string com::com( string module_name [, string server_name [, int codepage]])
```

参数说明：

module_name：被请求组件的名字或 class-id。

server_name：DCOM 服务器的名字。

codepage：指定用于将 PHP 字符串转换成 UNICODE 字符串的代码页，反之亦然。该参数的取值有 CP_ACP、CP_MACCP、CP_OEMCP、CP_SYMBOL、CP_THREAD_ACP、CP_UTF7 和 CP_UTF8。

设计过程

（1）建立数据库连接文件 conn.php，实现与 Access 数据库的连接，代码如下：

```
<?php
	$conn = new com("adodb.connection");
	$connstr="driver={microsoft access driver (*.mdb)}; dbq=". realpath("data/db_database11.mdb");
	$conn->open($connstr);
?>
```

（2）建立如图 11.1 所示的页面，并循环显示数据库中的所有图书信息，代码如下：

```
<?php
   $sql="select * from tb_book";
   $rs=new com("adodb.recordset");
   $rs->open($sql,$conn,1,3);
   while(!$rs->eof)
   {
  ?>
      <tr>
        <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo iconv('gbk','utf-8',$rs->fields(bookname)->value);?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo iconv('gbk','utf-8',$rs->fields(pub)->value);?></div></td>
      </tr>
```

密码，单击“确定”按钮即可。

11.2 操作 Access 数据库

Access 数据库是小型项目后台支持的首选，在相关的数据库操作上与其他数据库有所不同。本节将通过实例详细介绍如何操作 Access 数据库。

实例 508 通用数据信息的查询

光盘位置：光盘\MR\11\508

高级

趣味指数：★★★★

实例说明

本实例主要实现 Access 数据库信息的查询。运行本实例，在文本框中输入某学生的学号，如果查询到该学生的成绩将该学生的详细成绩信息显示出来，如图 11.4 所示。

成绩查询

请输入学号：0312101 查询

学号	姓名	班级	科目	成绩
0312101	小刘	三年一班	语文	10
0312101	小刘	三年二班	代数	22

不及格科目：2门

图 11.4 通用数据信息查询

关键技术

通过 ADO 方式建立与 Access 数据库的连接已经在前面介绍过，但建立完该连接后并不能对数据库进行查询、更新等操作，必须先实例化 COM 类建立记录集，然后通过 open()方法执行查询语句，其关键代码如下：

```
include("conn.php");                                                    //包含数据库连接文件
$rs=new com("adodb.recordset");                                         //建立记录集
$rs->open("select * from tb_score where sno='".$_POST[sno]."'",$conn,3,1);    //执行查询
```

设计过程

（1）新建 conn.php 文件，建立与 Access 数据库的连接，代码如下：

```
<?php
$conn=new com("adodb.connection");
$connstr="provider=microsoft.jet.oledb.4.0;data source=".realpath("../data/db_database11.mdb");
$conn->open($connstr);
?>
```

（2）新建 index.php 文件，创建 form 表单，将学号提交到本页，在本页中根据 POST 方法提交的学号执行查询操作，查询出符合条件的数据并且输出查询结果，代码如下：

```
<?php
   if($_POST[submit]!=""){
       include("conn.php");
         $rs=new com("adodb.recordset");
         $rs->open("select * from tb_score where sno='".$_POST[sno]."'",$conn,3,1);
         if($rs->eof|| $rs->bof){
         ?>
        <tr>
          <td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">没有查找到该学生成绩！</div></td>
          </tr>
         <?php
```

实例 509 查询结果的分页显示

光盘位置：光盘\MR\11\509

高级

趣味指数：★★★★

实例说明

如果记录集中的记录较多，用一个页面显示所有记录会给用户的浏览带来诸多不便。为了解决这个问题，可以对数据库中的数据进行分页显示。运行本实例，如图 11.5 所示。

图 11.5　数据的分页显示

关键技术

PHP 利用 COM 类连接数据库，可以直接使用类中的数据成员完成很多操作，常用的数据成员如表 11.3 所示。

表 11.3　记录集对象常用数据成员

数 据 成 员	说　　明
pagesize	设置分页显示时，记录集对象一页所容纳的记录数
absolutepage	设置当前显示的为第几页，首先应设置 pagesize 的值
bof	检查当前记录集对象所指位置是否为第一条记录之前
eof	检查当前记录集对象所指位置是否为最后一条记录之后
pagecount	显示记录集当前页面总数，首先应设置 pageesize 的值
movenext	将记录集对象的指针向下移动一条

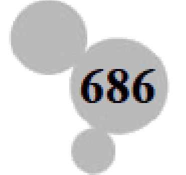
设计过程

（1）建立数据库连接文件 conn.php，代码如下：

```
<?php
	$conn = new com("adodb.connection");
	$connstr="driver={microsoft access driver (*.mdb)}; dbq=".
	realpath("../data/db_database11.mdb ");
	$conn->open($connstr);
?>
```

（2）根据接收到的查询字符串的值确定显示的信息页，代码如下：

```
<?php
	include("Conn/conn.php");                              //包含数据库连接文件 conn.php
	$sql="select * from tb_bookinfo order by pdate desc";  //查询 tb_bookinfo 表中的所有商品
	$rs=new com("adodb.recordset");                        //用 com 类声明一个记录集对象
	$rs->open($sql,$conn,1,3);                             //利用 com 类的 open()方法执行查询
	$rs->pagesize=20;                                      //设置每页最多显示 20 条记录
```



```
    if((trim(intval($_GET[page]))=="")||(intval($_GET[page])>$rs->pagecount)||(intval($_GET[page])<=0)) {
        $page=1;
    } else {                          //否则使$page 的值为所接收的查询字符串的数值
        $page=intval($_GET[page]);
    }
    if($rs->eof || $rs->bof) {        //判断表中的内容是否为空，如果为空则给出提示
?>
    <tr>
    <td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">本站暂无商品！
    </div></td>
    </tr>
    <?php
    }else{                            //不为空则分页显示数据
        $rs->absolutepage=$page;      //设置当前显示页为所接收的查询字符串的值
        $mypagesize=$rs->pagesize;    //定义变量$mypagesize 用于控制当前页的循环终止
      while(!$rs->eof && $mypagesize>0) {  //记录集对象没有到整个表尾并且当前页所要显示的记录没有完全显示完，则循环显示数据信息
 ?>
…
<?php
      $mypagesize--;                  //每显示一条记录，应使$mypagesize 的值减 1
      $rs->movenext;                  //使记录集指针移到下一条记录的位置
     }
}
?>
```

（3）向信息显示页面发送查询字符串从而实现商品信息的分页显示，代码如下：

```
      <div align="right">
<?php
    if($page>=2){
?>
  <a href="index.php?page=1" title="首页"><font face="webdings"> 9 </font></a>
  <a href="index.php?page=<?php echo $page-1;?>" title="前一页"><font face="webdings"> 7         </font></a>
<?php
      }
      if($rs->pagecount<=4){
            for($i=1;$i<=$rs->pagecount;$i++){
?>
              <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php
            }
       }else{
        for($i=1;$i<=4;$i++){
?>
               <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php
        }
?>
         <a href="index.php?page=<?php
        if($rs->pagecount>=$page+1)
             echo $page+1;
         else
             echo 1;
        ?>" title="后一页"><font face="webdings"> 8 </font></a>
       <a href="index.php?page=<?php echo $rs->pagecount;?>" title="尾页"><font face="webdings"> : </font></a>
       <?php
            }
        ?>
</div>
```

秘笈心法

心法领悟 509：比较 ASP 与 PHP。

熟悉 ASP 的读者可能会发现，PHP 利用 COM 类连接 Access 数据库的思路与 ASP 用 ADO 方法连接数据库的思路类似，同样实现翻页的思路也是类似的。

实例 510	用户身份验证类 光盘位置：光盘\MR\11\510	高级 趣味指数：★★★★

实例说明

利用用户身份验证类实现登录用户的身份验证，不仅为以后的维护工作带来很大方便，而且还可以提高程序的安全性。运行本实例，如图 11.6 所示，如果登录用户输入正确的用户名和密码则系统给出成功登录信息，反之系统给出登录失败信息。



图 11.6　登录 Access 数据库成功

关键技术

本实例主要通过面向对象的类来实现用户身份的验证，用户身份验证类的代码如下：

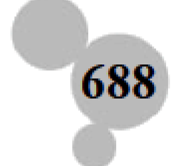
```
class chk{
        private $name;
         private $pwd;
         public function    construct($x,$y){
              $this->name=$x;
              $this->pwd=$y;
         }
        public function chkuser(){
               include_once("conn.php");
               $rs=new com("adodb.recordset");              //创建记录集对象
               $rs->open("select * from tb_user where username='".$this->name."' and userpwd='".$this->pwd."'",$conn,3,1);
                  if($rs->eof || $rs->bof){
                       echo "<script>alert('对不起，密码或用户名错误!');history.back();</script>";
                       exit;
                   }else{
                       echo "<script>alert('恭喜您登录成功!');history.back();</script>";
                       exit;
                   }
        }
  }
```

该类通过判断数据库中查询结果的游标位置来判断用户输入的用户名和密码是否正确，如果该游标位于第一条记录之前或最后一条记录之后，则说明用户输入的用户名或密码错误。

设计过程

（1）建立如图 11.6 所示的登录表单，代码如下：

```
   <form name="form1" method="post" action="index.php">
    <tr>
            <td height="25" colspan="2" bgcolor="#0099CC"><div align="center" class="STYLE1">用户身份验证</div></td>
         </tr>
         <tr>
            <td width="60" height="25" bgcolor="#FFFFFF"><div align="center">用户名：</div></td>
            <td width="187" bgcolor="#FFFFFF"><div align="left"> <input type="text" name="username" size="22"
class="inputcss"></div></td>
         </tr>
         <tr>
            <td height="25" bgcolor="#FFFFFF"><div align="center">密码：</div></td>
```



```
            <td height="25" bgcolor="#FFFFFF"><div align="left"> <input type="password" name="userpwd" size="22"
class="inputcss"></div></td>
         </tr>
    <tr>
            <td height="25" colspan="2" bgcolor="#FFFFFF"><div align="center"><input name="submit" type="submit" value="登录"
class="buttoncss"></div></td>
            </tr>
</form>
```

（2）建立密码验证类。

（3）类的实例化并实现用户身份的验证，代码如下：

```
$chk1=new chk($username,$userpwd);
$chk1->chkuser();
```

秘笈心法

心法领悟 510：类的实例化。

一般情况下，创建一个类后并不能直接对类中的属性和方法进行引用，必须对类进行实例化，即创建一个对象。PHP 中用 new 关键字创建对象。

实例 511　图片的分栏显示

光盘位置：光盘\MR\11\511

高级

趣味指数：★★★★

实例说明

采用分栏的方式将多张图片同时显示在一个页面中，可以使页面更加整齐。运行本实例，效果如图 11.7 所示。

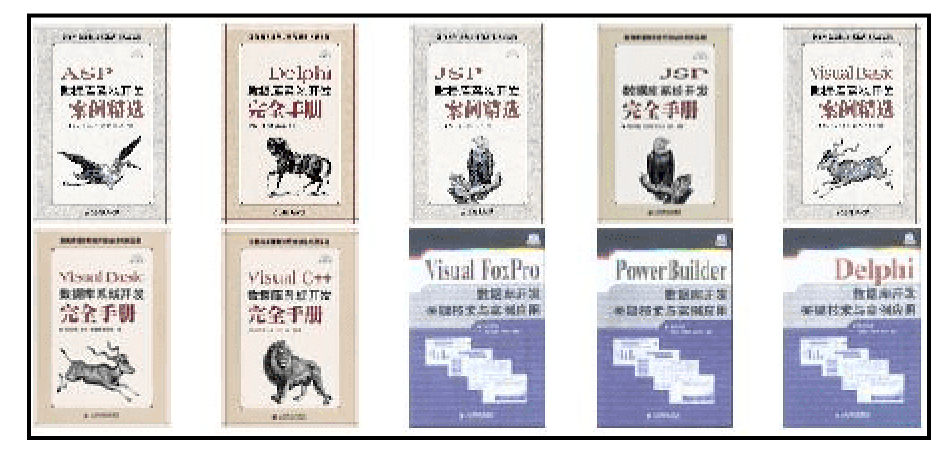


图 11.7　图片的分栏显示

关键技术

本实例实现图片的分栏显示主要依靠标记变量$i，如果该变量的值能被 5 整除则显示下一行（当然这是一栏显示 5 张图片的情况，如果一栏显示 4 张图片，可以通过判断该值能否被 4 整除，依此类推）。注意，显示一张图片后应使$i 的数值加 1，实现该过程的代码如下：

```
<?php
  $i=1;
        while(!$rs->eof){
                if ($i % 5==0){
        ?>
            <td height="25">
               <div align="left"><img src="<?php echo iconv('gbk','utf-8',$rs->fields(address)->value);?>" border="0"> </div>
          </td>
          </tr>
          <?php
                 }else{
       ?>
  <td height="25"><div align="left"><img src="<?php echo iconv('gbk','utf-8',$rs->fields(address)->value);?>" border="0"> </div>
  </td>
       <?php
             }
          $rs->movenext;
          $i++;
       }
 ?>
```

设计过程

（1）建立与 Access 数据库的连接，代码如下：

（2）编写“显示信息”按钮，单击此按钮，将输出 Access 数据库中存储的所有数据，代码如下：

```
<?php
if($_GET[id] == "1"){                                                    //单击按钮
	$rst = $conn -> execute("select * from tb_demo") or die ("出错");       //返回结果集
?>
<table bgcolor="#80E7EC" width="580px"><tr><td align="center">图书 ID</td><td align="center">图书名称</td><td align="center">作者</td><td
align="center">出版社</td><td align="center">出版时间</td></tr>
<?php
	while(!$rst -> EOF){                                                 //如果指针没有指向最后一条数据
?>
<tr>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[0]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[1]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[2]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[3]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[4]);?></td>
</tr>
<?php
		$rst -> movenext();                                              //指针向下移动
	}
?>
```

（3）编写表单，通过字符串拆分函数，对文本框内容进行拆分并执行插入操作，代码如下：

```
<?php
if($_POST[sub]){                                                                              //单击按钮
	if($_POST[te] == "" || $_POST[te] == "输入字段信息每一项用空格分割"){                          //判断文本框内容
		echo "<script>alert('请在文本框中输入内容');location.href='index.php?id=1';</script>";     //JavaScript 提示
	}else{
		$array = explode(" ",$_POST[te]);                                                     //拆分字符串
		if(count($array) != 5){                                                               //判断数组
			echo "<script>alert('文本框输入内容有误,数据表共 5 个字段');location.href='index.php?id=1';</script>";   ///JavaScript 提示
		}else{
			$a = iconv('utf-8','gbk',$array[0]);                                              //转码操作
			$b = iconv('utf-8','gbk',$array[1]);
			$c = iconv('utf-8','gbk',$array[2]);
			$d = iconv('utf-8','gbk',$array[3]);
			$e = iconv('utf-8','gbk',$array[4]);
			//判断是否插入成功
			if($conn -> execute("insert into tb_demo(`id`,`bname`,`author`,`pub`,`date`)values('".$a."','".$b."','".$c."','".$d."','".$e."')")){
				echo "<script>alert('插入数据成功');location.href='index.php?id=1';</script>";   //JavaScript 提示
			}else{
				echo "<script>alert('SQL 语句有误，请重新输入');location.href='index.php?id=1';</script>";
			}
		}
	}
}
?>
```

秘笈心法

心法领悟 512：Access 数据库的 SQL 语句。

Access 数据库与 MySQL 数据库的 SQL 语句在更新、删除、查询方面几乎相同，但是在插入语句方面有所不同。在编辑插入的 insert 语句时，数据表字段必须由“`”包含。

实例 513	删除图书信息 光盘位置：光盘\MR\11\513	高级 趣味指数：★★★★

实例说明

Access 数据库虽然小巧，但是在功能上并不逊于其他数据库，进行简单的插入、删除操作是很迅速的。本实例通过 ADODB 连接 Access 数据库实现删除图书信息的操作，运行结果如图 11.9 所示。

ADODB，全称为 Active Data Objects Data Base，是存取数据库所使用到的一组函数。虽然 PHP 是构建 Web 系统强有力的工具，但是由于 PHP 存取函数一直没有标准化，不同数据库间的函数名称、参数差异很大，在更换数据库时会带来大量的代码修复工作。这时就需要一组函数库来隐藏不同数据库函数间的差异，以便开发者可以简单地切换数据库，这就是 ADODB 类库，其详细的讲解请读者参考本书第 13 章。

11.3　Access 数据库的安全

数据库文件的安全非常重要，尤其是对于利用 Access 数据库建立的小型网站而言，用户总是担心数据库文件会不会被下载、用户名和密码是否会被他人盗走、系统是否会被人攻击等。下面通过具体实例来讲解网站的安全措施。

实例 514　操作带密码的 Access 数据库　高级

光盘位置：光盘\MR\11\514　趣味指数：★★★★

实例说明

为防止用户通过下载 Access 数据库文件来达到窃取机密信息的目的，可以为 Access 数据库设置密码，以保护数据，这样即使用户得到数据库也会因为没有密码而无法查看数据库中的内容。运行本实例，对 Access 数据库设置密码的效果如图 11.10 所示，在网站中连接数据库的运行结果如图 11.11 所示。

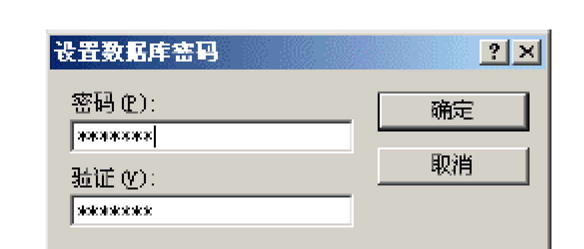


图 11.10　设置数据库密码



书名	出版社	出版时间	页数	价格
Delphi数据库开发关键技术与实例应用	人民邮电出版社	2004年5月	407 页	40 元
Delphi数据库开发实例解析	机械工业出版社	2004年2月	480 页	49 元
ASP数据库开发实例解析	机械工业出版社	2004年12月	321 页	36 元
Visual FoxPro数据库开发实例解析	机械工业出版社	2003年9月	458 页	48 元
Power Builder数据库开发实例解析	机械工业出版社	2003年9月	400 页	43 元
Visual Basic数据库开发实例解析	机械工业出版社	2003年8月	460 页	48 元
Visual Basic精彩编程200例	机械工业出版社	2003年1月	358 页	39 元

本站共有图书47种 每页显示10种 第5页/共5页　1 2 3 4 GO

图 11.11　操作带密码的 Access 数据库

关键技术

本实例主要通过 Access 菜单栏中的“工具”/“安全”/“设置数据库密码”命令来设置 Access 数据库的密码，同时也可以撤销数据库密码。

撤销数据库密码的过程为选择 Access 菜单栏中的“工具”/“安全”/“撤销数据库密码”命令，在弹出的“撤销数据库密码”对话框中输入设置时的密码即可。

提示：*在设置或者撤销 Access 数据库的密码时，必须将数据库以独占的方式打开。*

设计过程

（1）启动 Access 数据库，单击工具栏中的按钮，弹出“打开”对话框，选择需要设置密码的数据库文件，例如 bookinfo.mdb，单击“打开”按钮右侧的下三角按钮，在弹出的下拉列表中选择“以独占方式打开”选项，则该数据库文件将以独占方式打开。

设计过程

（1）修改实例中所用数据库的扩展名，将其命名为 db_news.php。

（2）新建 conn 文件夹，编辑 conn.php 文件，定义 Access 数据库的连接方法，其代码如下：

```
<?php
$conn = new com("adodb.connection");
$connstr="driver={microsoft access driver (*.mdb)}; dbq=". realpath("data/db_news.php");
$conn->open($connstr);
?>
```

（3）新建 index.php 文件，包含数据库连接文件，应用 while 语句完成数据库中数据的循环输出，其关键代码如下：

```
<?php
    include("conn/conn.php");                                    //包含数据库连接文件
    $sql="select * from tb_news order by pdate desc";            //定义查询语句
    $rs=new com("adodb.recordset");                              //实例化类
    $rs->open($sql,$conn,1,3);                                   //执行查询操作
    if($rs->eof || $rs->bof){                                    //判断查询结果是否为真
?>
<tr><td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">暂无新闻信息！</div></td></tr>
<?php
        }else{
          while(!$rs->eof){                                      //循环输出查询结果
  ?>
<tr>
    <td height="20" align="center" bgcolor="#FFFFFF"><div align="center">
        <?php $fields=$rs->fields(type);echo $fields->value;?>
                </div></td>
    <td height="20" align="left" bgcolor="#FFFFFF"><div align="left">
        <?php $fields=$rs->fields(newobj);echo $fields->value;?>
    </div></td>
</tr>
<?php
        $rs->movenext;
        }
    }
?>
```

秘笈心法

心法领悟 515：Access 数据库的安全。

首先为 Access 数据库设置密码，然后更改 Access 数据库文件的后缀名称，通过这两项设置，相信 Access 数据库的安全可以得到一定的保障。

11.4　Access 数据库的综合运用

本节以 Access 数据库为数据的存储方式，开发一个聊天室功能模块，通过这个功能模块体会 Access 数据库在完整项目中的运用。

实例 516	ADODB 连接 Access 数据库 光盘位置：光盘\MR\11\516	高级 趣味指数：★★★★

实例说明

在本实例中介绍通过 ADODB 类库连接 Access 数据库，这里连接的 Access 数据库是具有密码的，并且隐藏了 Access 数据库文件的后缀。运行本模块中的 top.php 文件，就可以测试出数据库是否连接成功，如果连接



成功，将输出如图 11.14 所示的页面。

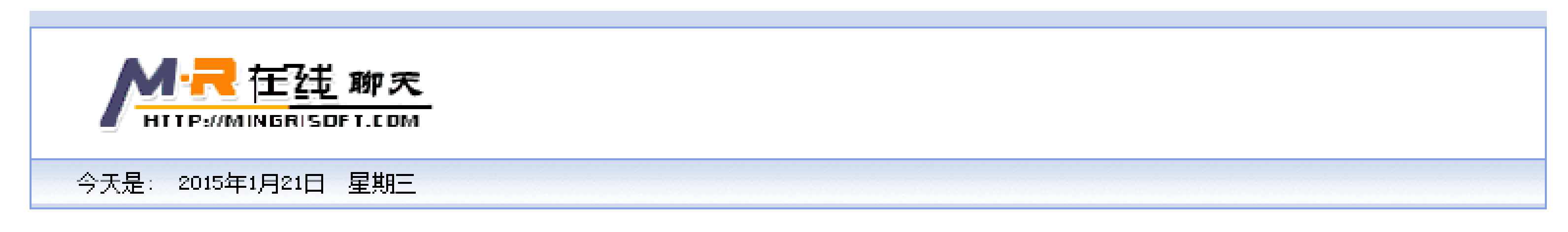


图 11.14　成功连接 Access 数据库

关键技术

这里应用 ADODB 类库连接 Access 数据库，并且 Access 数据库设置了密码，同时对数据库文件名称的后缀进行了修改。所以在封装 Access 数据库的连接类时，必须注意以下内容：

☑　要指定正确的 ADODB 类库的调用路径。

☑　由于使用 ADODB 类库操作 Access 数据库，数据库类型的判断应该使用 ado_access。

☑　定义连接语句时，不要忘记 Driver 与(*.mdb)之间的空格。

☑　Access 数据库的访问密码为 mrsoft。

☑　Access 数据库文件的后缀是.php。

设计过程

（1）定位 ADODB 类库的位置，将其存储于章目录 11 下，命名为 adodb5。

（2）定位 Access 数据库文件的位置，将其存储于 11\011\data 文件夹下，命名为 db_messages.php。

（3）在 11\011\conn 文件夹下新建 ConnDB.php 文件，封装 ADODB 连接 Access 数据库类，其代码如下：

```
<?php
class ConnDB{
    var $dbType;                                          //数据库类型标识
    var $host;                                            //数据库服务器地址
    var $userName;                                        //数据库服务器用户名
    var $pwd;                                             //数据库用户密码
    var $dbName;                                          //要连接的数据库名
    var $isDebug;                                         //是否显示调试信息
    var $connID;                                          //数据库连接 ID
    function ConnDB ($dbType = 'mysql', $host, $userName, $pwd, $dbName, $isDebug = false){
        $this->dbType = $dbType;                          //为数据库类型赋初值
        $this->host = $host;                              //为服务器地址赋初值
        $this->userName = $userName;                      //为用户名赋初值
        $this->pwd = $pwd;                                //为密码赋初值
        $this->dbName = $dbName;                          //为数据库名赋初值
        $this->isDebug = $isDebug;                        //为调试信息赋初值
    }
    function getConnID (){
        require_once '../adodb5/adodb.inc.php';           //导入 ADODB 类库
        $this->connID = NewADOConnection($this->dbType);  //生成数据库连接对象
        if ($this->dbType == 'mysql' || $this->dbType == 'mssql') { //如果为 MySQL 数据库或者 SQL Server 数据库
            $this->connID->Connect($this->host, $this->userName, $this->pwd, $this->dbName);    //通过 Connect()方法建立与数据库的连接
            if ($this->dbType == 'mysql') {
                $this->connID->Execute('set names gb2312');    //设置数据库编码
            }
        } elseif ($this->dbType == 'ado_access') {        //如果为 Access 数据库
            //在连接 Access 数据库时，编写连接语句一定要注意空格的使用，必须严格执行，否则很可能导致连接失败
            $this->connID->Connect('Driver={Microsoft Access Driver (*.mdb)};Uid='.$this->userName.'; Pwd='.$this->pwd .';
Dbq='.realpath($this->dbName).';');
        } else {
            return false;
        }
        $this->connID->debug = $this->isDebug;            //是否调试
        return $this->connID;                             //返回数据库连接对象
    }
```

```
        function closeConnID (){
            @$this->connID->Disconnect();                           //关闭与数据库的连接
        }
}
```

（4）如果要通过封装的 ConnDB 类连接数据库，必须对类进行实例化，同时向类中传递参数，最终返回连接标识，该操作被存储于 config.inc.php 文件中。在该文件中完成数据库连接类的实例化，返回连接的标识，同时数据库操作类的实例化也在该文件中完成，其代码如下：

```
<?php
session_start();                                                              //初始化 SESSION 变量
require_once 'ConnDB.php';                                                    //包含数据库连接类文件
require_once 'AdminDB.php';                                                   //包含数据库操作类文件
//$connDB = new ConnDB('access','','','data/db_messages.php',false);          //连接没有密码的 Access 数据库
$connDB = new ConnDB('ado_access','','mrsoft','data/db_messages.php',false);  //连接 Access 数据库
$conn = $connDB->getConnID();                                                 //返回连接标识
$adminDB = new AdminDB();                                                     //实例化数据库操作类
```

（5）新建聊天室头文件 top.php，通过 include_once 语句包含 config.inc.php 文件，返回数据库的连接标识和数据库操作类的实例化对象，在该文件中判断当前用户的权限，根据不同的权限输出不同的内容，其关键代码如下：

```
<?php
include_once("conn/config.inc.php");            //包含实例化文件
?>
<?php
if($_SESSION["nc"]!=""){
        $arr=$adminDB->executeSQL("select * from tb_puser where usernc='".$_SESSION["nc"]."'",$conn);
        if($arr[0][5]==1){
?>
|          <a href="setwordnull.php">清空聊天记录</a> |
<?php
        }
}
?>
<?php
    if($_SESSION["nc"]!=""){
?>
            | <a href="logout.php">退出聊天室</a> |
<?php
    }
?>
```

秘笈心法

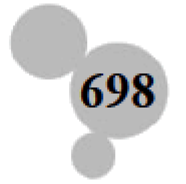
心法领悟 516：通过 ADODB 类库操作具有密码的 Access 数据库。

在通过 ADODB 类库操作 Access 数据库时，如果 Access 数据库没有设置密码，那么设置 Access 数据库的类型为 access；如果 Access 数据库设置了密码，那么 Access 数据库的类型必须设置为 ado_access。

实例 517	ADODB 操作 Access 数据库 光盘位置：光盘\MR\11\517	高级 趣味指数：★★★★

实例说明

在本实例中讲解 ADODB 操作 Access 数据库方法的封装。在聊天室的用户登录模块中，应用数据库操作 AdminDB 中的 executeSQL()方法完成登录用户的验证和登录信息的添加操作，其运行效果如图 11.15 所示。



（2）在 js/check.js 文件的 open_chknc()方法中调用 chkusernc.php 文件，验证提交的用户名是否被占用，其关键代码如下：

```
<?php
$nc=trim($_GET[nc]);
if($nc==""){
	echo "请输入用户昵称!";
}else{
	$info=$adminDB->executeSQL("select * from tb_puser where usernc='".$nc."'",$conn);
	if(!$info){
		echo "恭喜您,该昵称未被占用!";
	}else{
		echo "对不起,该昵称已被占用!";
	}
}
?>
```

（3）新建 saveuser.php 文件，验证登录用户名是否被占用，如果用户名可用，将用户登录信息存储到指定数据表中，并且跳转到聊天室主页面，否则提示用户名不可用，或者用户登录失败，其关键代码如下：

```
<?php
include_once("conn/config.inc.php");                                       //包含数据库连接、操作类的实例化文件
$usernc=trim($_POST[usernc]);                                              //获取提交的用户名
$sql=$adminDB->executeSQL("select * from tb_puser where usernc='".$usernc."'",$conn);    //验证用户名是否被占用
if($sql){
  echo "<script language='javascript'>alert('对不起，该昵称已被其他用户使用!');history.back();</script>";
  exit;
}else{
  $logintime=date("Y-m-j H:i:s");                                          //定义时间
  $userhead=$_POST[userhead];
  $online=1;
  $ip=getenv("REMOTE_ADDR");                                               //获取 IP
  $email=trim($_POST[email]);                                              //获取用户提交的 E-mail
  $mark=0;
  $nowtime=time();                                                         //定义时间戳
  $arr=$adminDB->executeSQL("insert into tb_puser(usernc,logintime,ip,email,mark,userhead,online,nowtime) values ('$usernc','$logintime','$ip',
'$email','$mark','$userhead','$online','$nowtime')",$conn);                //执行用户信息的添加操作
  if($arr){
      $_SESSION["nc"]=$usernc;                                             //用户登录成功后将用户名赋给 SESSION 变量
      echo "<script>window.location.href='chat.php';</script>";            //跳转到聊天室主页
  }else{
      echo "添加失败！ ";
  }
}
?>
```

秘笈心法

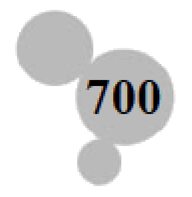
心法领悟 517：使用 getenv()函数获取客户端的 IP 地址。

在本实例中，应用 getenv()函数获取客户端的 IP 地址，其作用与$_SERVER['REMOTE_ADDR']是相同的。

实例 518	聊天室 光盘位置：光盘\MR\11\518	高级 趣味指数：★★★★

实例说明

聊天室的主页面由头文件 top.php、尾文件 bottom.php、发布聊天信息 chat.php、聊天内容 listchat.php 和聊天参与者 listuser.php 共 5 个部分组成，其运行效果如图 11.16 所示。



```
                        } else {
                                echo $_SESSION ["snc"];
                        }
                }
        }
        ?>">
      <input type="hidden" name="touser1"value="
<?php
        $usernc = trim ( $_SESSION ["nc"] );
        $info = $adminDB->executeSQL ( "select * from tb_puser where usernc='" . $usernc . "'", $conn );
        if ($info [0] [5] == 1) {
                if ($_SESSION ["snc"] == "") {
                        echo "所有人";
                } else {
                        echo $_SESSION ["snc"];
                }
        } else {
                if ($infoc [total] == 0) {
                        echo "请选择发送人";
                } else {
                        if ($_SESSION ["snc"] == "") {
                                echo "请选择发送人";
                        } else {
                                echo $_SESSION ["snc"];
                        }
                }
        }
?>">   
        <input type="submit" name="submit" value="发送" class="buttoncss" />  
        <input name="reset" type="reset" class="buttoncss" value="重写">
</form>
```

（2）新建 saveword.php 文件，将聊天信息存储到指定的数据表中，其代码如下：

```
<?php
include_once ("conn/config.inc.php");                                                    //包含配置文件
$fromuser = trim ( $_SESSION ["nc"] );                                                   //获取登录用户名称
$sql = $adminDB->executeSQL ( "select * from tb_puser where usernc='" . $fromuser . "'", $conn );     //判断用户是否存在
if (! $sql) {
        echo "<script>alert('对不起,等待时间过长，请重新登录!');window.location.href='logout.php';</script>";
        exit ();
}
$touser = $_POST ['touser1'];
if ($touser == "所有人") {
        $touser = "all";
}
$content = $_POST ['bbs_content'];
$pubtime = date ( "Y-m-j H:i:s" );
$nowtime = time ();
$arr = $adminDB->executeSQL ( "insert into tb_message(touser,fromuser,content,pubtime,nowtime) values ('$touser','$fromuser', '$content', '$pubtime',
'$nowtime')", $conn );                                                                   //执行信息的添加操作
if ($arr) {
        echo "<script>window.location.href='chat.php';</script>";
} else {
        echo "添加失败！";
}
?>
```

（3）新建 listuser.php 文件，输出聊天室当前的在线用户，并通过 JavaScript 脚本中的 selectuser()方法调用 selectuser.php 文件，完成聊天对象的选择，其关键代码如下：

```
<script language="javascript">
function selectuser(x){
      parent.form_bbs.touser.value=x;
      parent.form_bbs.touser1.value=x;
      window.location.href='selectuser.php?usernc='+x;
}
</script>
<table width="120" border="0" align="center" cellpadding="0" cellspacing="0">
  <?php
 $array = $adminDB->executeSQL ( "select * from tb_puser where online=1 ", $conn );
```



```
        }
}
?>
</table>
```

（6）新建 function.php 文件，编辑自定义函数 unhtml()对聊天室中输出的聊天信息进行格式化；编辑 getday()函数计算时间。

秘笈心法

心法领悟 518：私聊功能的实现。

PHP 中实现私聊的方法有多种，用 SESSION 实现私聊相对简单，本聊天室中私聊的关键代码如下：

```
<?php
  if($_SESSION["nc"]==$info[touser] ||$_SESSION["nc"]==$info[fromuser] ){
?>
  ... //显示私聊内容
<?php
    }
?>
```

上述简短的代码说明了实现私聊的原理，$_SESSION["nc"]保存的是当前登录用户的昵称，$info[touser]表示当前聊天内容要发送的对象。如果两者相等，则显示出该聊天内容；否则不显示，或者在$_SESSION["nc"]==$info[fromuser]的情况下也显示聊天内容，这种情况表示发送用户可以看到自己所发送的聊天内容。

实例 519	聊天室管理 光盘位置：光盘\MR\11\519	高级 趣味指数：★★★★

实例说明

在聊天室模块中有一个特殊用户 tsoft，即聊天室的管理者，可以行使聊天室的管理权限，发布公告消息、清空聊天记录。管理员登录的运行效果如图 11.17 所示。



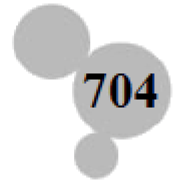
图 11.17　聊天室管理员登录

关键技术

在聊天室管理员登录页面中，关键是通过 AdminDB 类中的 executeSQL()方法判断输入的用户名和密码是否正确，如果正确则说明是管理员，将该用户名存储到 SESSION 变量中。只有当前用户具备了管理员的权限才可以执行清空聊天记录的操作。

设计过程

（1）新建 admin_login.php 文件，创建 form 表单将用户名和密码提交到 admin_login_ok.php 文件中，完成管理员登录操作。



12.1　连接 Oracle 数据库

本节将介绍在 PHP 中如何实现与 Oracle 数据库的连接，这里分别讲解通过 Oracle 函数库和 ADODB 连接及操作 Oracle 数据库。

实例 520　oci_connect()函数连接 Oracle 数据库

光盘位置：光盘\MR\12\520

高级

趣味指数：★★★★☆

实例说明

PHP 连接 Oracle 数据库的方法很多，这里介绍通过 PHP 中自带的 Oracle 函数库来完成对 Oracle 数据库的操作。在本实例中，应用 oci_connect()函数连接 Oracle 数据库，然后应用 oci_parse()函数执行查询语句，最终通过 oci_fetch_array()函数和 while 语句完成数据库中数据的循环输出，其运行结果如图 12.1 所示。



图 12.1　oci_connect()函数连接 Oracle 数据库

关键技术

在 PHP 的 Oracle 函数库中提供有完成对 Oracle 数据库操作的函数，这里只对本实例中应用的函数进行详细介绍。

（1）oci_connect()函数建立一个到 Oracle 服务器的连接，返回一个连接标识符，其语法如下：

```
resource oci_connect( string username, string password [, string db [, string charset [, int session_mode]]] )
```

oci_connect()函数的参数说明如表 12.1 所示。

表 12.1　oci_connect()函数的参数说明

参　数	说　明
username	必选参数，指定数据库中的用户名
password	必选参数，指定数据库中用户的密码
db	可选参数，包含要连接的本地 Oracle 的实例或者在 tnsnames.ora 中条目的名字。如果没有指定该参数，PHP 使用环境变量 ORACLE_SID（Oracle 实例）或 TWO_TASK（tnsnames.ora）来确定连接哪一个数据库
charset	使用 Oracle 服务器 9.2 或更高版本时，可以指定 charset 参数，被用于新的连接。如果使用低于 9.2 版本的 Oracle 服务器，本参数将被忽略并以 NLS_LANG 环境变量替代
session_mode	该参数自版本 1.1 起可用并接受如下值：OCI_DEFAULT、OCI_SYSOPER 和 OCI_SYSDBA。如果指定 OCI_SYSOPER 或 OCI_SYSDBA 其中之一，oci_connect()函数将尝试使用外部认证信息建立特权连接。特权连接默认被禁止，如果要启用，需要将 oci8.privileged_connect 设置为 On

说明： 对 oci_connect()函数用同样的参数第二次及之后的调用将返回第一次调用所返回的连接句柄。即对一个句柄发出的查询也适用于另一个句柄，因为它们是同一个句柄。如果需要两个句柄在事务上互相隔离，则使用 oci_new_connect()函数替代。

（2）oci_parse()函数配置 Oracle 语句预备执行，其语法如下：

```
resource oci_parse( resource connection, string query )
```

参数说明：

connection：必选参数，数据库连接成功后返回的连接标识符。

query：必选参数，执行的 SQL 语句。

oci_parse()函数是最常用的函数，在操作 Oracle 数据库时，无论是执行查询、修改、删除或更新操作，该函数是必须使用的，只有通过该函数在 connection 上配置 query 并返回语句标识符后，才可以用于其他函数的操作，例如 oci_bind_by_name()函数、oci_execute()函数。oci_parse()函数在出错时返回 FALSE。

技巧： oci_parse()函数并不验证参数 query。验证参数 query 是否为合法的 SQL 或者 PL/SQL 语句的唯一方法是执行语句。

（3）oci_execute()函数执行一条之前被解析过的语句（参见 oci_parse()函数）。如果成功则返回 TRUE；否则返回 FALSE，其语法如下：

```
bool oci_execute ( resource stmt [, int mode] )
```

参数说明：

stmt：必选参数，获取的语句标识符。

mode：可选参数，mode 允许定义执行模式（默认是 OCI_COMMIT_ON_SUCCESS）。如果不需要将语句自动提交，则把 mode 设为 OCI_DEFAULT。

注意： 在使用 OCI_DEFAULT 模式时将建立一个事务，事务会在关闭连接或者脚本结束时自动回滚。需要明确调用 oci_commit()函数来提交事务，或者调用 oci_rollback()函数中止事务。

（4）oci_fetch_array()函数，从结果集中获取一行作为关联数组或数字数组或者二者兼有。如果没有更多行则返回 FALSE，其语法如下：

```
array oci_fetch_array( resource statement [, int mode] )
```

oci_fetch_array()函数中可选的第 2 个参数 mode 是一个常量，该参数的默认值是 OCI_BOTH，其可选值如表 12.2 所示。

表 12.2　oci_fetch_array()函数的参数 mode 的可选值

参　　数	说　　明
OCI_BOTH	得到一个同时包含关联和数字索引的数组
OCI_ASSOC	得到关联索引（如同 oci_fetch_assoc()）
OCI_NUM	得到数字索引（如同 oci_fetch_row()）
OCI_RETURN_NULLS	创建一个空元素为 NULL 字段
OCI_RETURN_LOBS	返回 LOB 的值的描述

oci_fetch_array()函数是 oci_fetch_row()函数的扩展版本。除将数据以数字索引方式存储在数组中之外，还可以将数据作为关联索引存储，用字段名作为键名。

如果结果中有两个或以上的列具有相同的字段名，最后一列将优先。要访问同名的其他列，必须用该列的数字索引或给该列起个别名。对于有别名的列，不能再用原来的列名访问其内容。

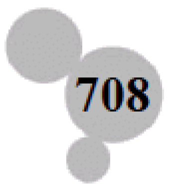




图 12.2　ADODB 操作 Oracle 数据库

说明：因为 ADODB 属于第三方组件，所以在本书的光盘中没有提供 adodb 类库文件，需要读者自行下载并复制到本章的 12 文件夹下，并且命名为 adodb5，然后才可以运行本章的实例。

关键技术

使用 ADODB 连接数据库系统应用 ADONewConnection()函数；如果选择持久化连接数据库则使用 PConnect()函数，如果选择非持久化连接则使用 Connect()函数。

（1）ADONewConnection()函数用于连接数据库系统，其语法如下：

```
ADONewConnection($databaseType)
```

参数$databaseType 表示要连接的数据库系统的名称，如 mysql、oci8 等。

（2）PConnect()函数实现与数据库持久化连接，其语法如下：

```
PConnect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

$database：使用到的数据库。

（3）Connect()函数，实现与数据库非持久化连接，其语法如下：

```
Connect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

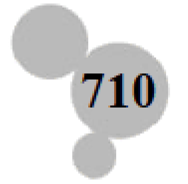
$database：使用到的数据库。

技巧：持久化连接和非持久化连接的区别：持久化连接不用每次都创建新连接，可以增加程序的执行速度，但是有些数据库不支持此函数，那么就需要使用 Connect()函数来替代 PConnect()函数。

（4）应用 execute()函数执行 SQL 语句，应用 EOF 变量、while 语句和 MoveNext()函数完成数据的循环输出。有关 ADODB 类库中函数的详细讲解请参考第 13 章。

设计过程

（1）将下载的 ADODB 类库命名为 adodb5，存储于 12 文件夹下，并且在 12 文件夹下创建文件夹 521 作



实例 522	删除 Oracle 数据库中的数据 光盘位置：光盘\MR\12\522	高级 趣味指数：★★★☆

实例说明

在本实例中，仍然应用 ADODB 操作 Oracle 数据库，在完成数据库中的数据循环输出的同时，增加一个数据库中数据的删除操作，通过超链接删除员工信息，其运行结果如图 12.3 所示。



图 12.3　删除 Oracle 数据库中的数据

关键技术

本实例同样应用 ADODB 类库完成对 Oracle 数据库的连接和操作，具体连接方法可以参考实例 521，这里不再赘述。

（1）execute()函数执行 SQL 语句，并返回一个结果集（ADORecordSet 对象），失败则返回 FALSE，其语法如下：

```
execute($sql[,$inputarr=false])
```

参数$sql 指定要执行的 SQL 语句；参数$inputarr 设置传入的结合变量，如果没有设置$inputarr，则$sql 为普通的 SQL 语句，那么 execute()函数的格式如下：

```
$connect->execute('select * from tb_user where id = 1')
```

如果设置了$inputarr，那么 execute()函数的格式如下：

```
$connect -> execute('select * from tb_user where id = ?',array($vl))
```

（2）要对结果集进行操作，首先要确定结果集中指针的位置，控制结果集中指针的方法如表 12.3 所示。

表 12.3　ADODB 类库控制结果集中指针的方法

方　法	说　明
fields 变量	保存当前指针所指向的记录
EOF 变量	记录当前指针是否指向最后一条记录。如果是，则返回 TRUE，否则返回 FALSE
MoveNext()	将 ADORecordSet（结果集）的指针下移一位。成功则返回 TRUE，否则返回 FALSE
Move($to)	将 ADORecordSet（结果集）的指针移动到指定位置。如果$to 等于 0，则指针指向结果集的第一条数据；如果$to 的值大于结果集，则指针指向最后一条数据。注意，这里的变量$to 只能是绝对定位
MoveFirst()	将指针移动到第一条数据，等同于 Move(0)
MoveLast()	将指针移动到最后一条数据，等同于 Move(RecordCount()-1)



设计过程

这里在实例 521 的基础上增加删除数据库中数据的功能，其具体的方法如下：

（1）在 index.php 文件中为输出的每条数据添加一个删除超链接，链接到 delete_ok.php 文件，将数据的 ID 作为参数传递到 delete_ok.php 文件中，其关键代码如下：

```
<a href="delete_ok.php?conn_id=<?php echo $rst->fields ['0']; ?>">删除</a>
```

（2）创建 delete_ok.php 文件，以超链接传递的参数值为条件，执行删除操作，删除数据表指定的记录，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );                    //设置文件编码格式
include("conn/conn.php");                                                //包含数据库连接文件
$sql="delete from tb_user where id='".$_GET['conn_id']."'";             //定义删除语句
$rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );   //执行删除语句
if($rst){
        echo "<script>alert('信息删除成功！');window.location.href='index.php';</script>";
}else{
        echo "<script>alert('信息删除失败！');windwo.location.href='index.php'</script>";
}
?>
```

秘笈心法

心法领悟 522：ADODB 类库。

在本章中只是应用 ADODB 类库连接及操作 Oracle 数据库，其实 ADODB 连接和操作任何数据库的函数、方法都是相同的，有关 ADODB 类库的具体运用将在第 13 章中进行详细的讲解。

12.2　操作 Oracle 数据库

PHP 对 Oracle 数据库有很好的函数支持，用户可以通过这些函数像操作 MySQL 数据库一样操作 Oracle 数据库。

实例 523	员工信息的添加、更新和查询 光盘位置：光盘\MR\12\523	高级 趣味指数：★★★☆

实例说明

本实例应用 PHP 和 Oracle 数据库创建一个企业用户管理模块，实现用户信息的添加、查询和更新操作。运行结果如图 12.4 所示。



图 12.4　企业用户管理操作页面

设计过程

本实例所讲解的技术已经在实例 523 中应用，在本实例中完全应用上一实例的内容，有关其创建过程这里不再赘述。

定义到 index.php 文件，在员工信息查询的 form1 表单中，员工 ID 下拉列表框值就是通过 oci_fetch_array() 函数和 while 语句动态添加的，其关键代码如下：

```
<form id="form1" name="form1" method="post" action="index.php">
  <tr>
    <td height="30" align="right" class="STYLE3">按照 ID 搜索用户:</td>
<td height="30" align="left" class="STYLE3">
<select name="select">
<?php
include("conn/conn.php");
$query="select * from tb_user ";
$result=oci_parse($conn,$query);
$r = oci_execute($result, OCI_DEFAULT);
while($row=oci_fetch_array($result,OCI_RETURN_NULLS)){
?>
      <option value="<?php echo $row[0];?>"><?php echo $row[0];?></option>
<?php
 }
oci_close($conn);
?>
    </select></td>
  </tr>
</form>
```

有关本实例的其他内容在实例 523 中已经介绍，这里不再赘述。

秘笈心法

心法领悟 524：使用 oci_fetch_array()函数的注意事项。

（1）本函数返回的字段名是区分大小写的。

（2）使用 oci_fetch_array()函数并不比使用 oci_fetch_row()函数慢，而且还能提供更多的返回值。

实例 525	统计数据库中数据的记录数 光盘位置：光盘\MR\12\525	高级 趣味指数：★★★☆

实例说明

在本实例中统计数据库中数据的记录总数并且循环输出数据库中的数据，其运行结果如图 12.6 所示。



图 12.6　统计数据库中数据的记录总数

关键技术

本实例中，统计数据库中数据的记录总数应用的是 oci_fetch_all()函数，然后在 foreach 和 for 语句的辅助之



下完成数据库中数据的循环输出。

oci_fetch_all()函数将获取的所有数据存储到一个数组中。它从一个结果中获取所有的行到一个用户定义的数组，返回获取的行数，如出错则返回 FALSE，其语法如下：

```
int oci_fetch_all ( resource statement, array &output [, int skip [, int maxrows [, int flags]]] )
```

oci_fetch_all()函数的参数说明如表 12.6 所示。

表 12.6　oci_fetch_all()函数的参数说明

参　数	说　明
statement	必选参数，获取的语句标识符
output	必选参数，定义存储数据的数组
skip	可选参数，从结果中获取数据时，最开始忽略的行数（默认值是 0，即从第一行开始）
maxrows	可选参数，要读取的行数，从第 skip 行开始（默认值是-1，即所有行）
flags	可选参数，flags 参数可以是下列值的任意组合： OCI_FETCHSTATEMENT_BY_ROW OCI_FETCHSTATEMENT_BY_COLUMN（默认值） OCI_NUM OCI_ASSOC

设计过程

（1）新建 conn 文件夹，编写 conn.php 文件连接 Oracle 数据库，并设置数据库的编码格式为 UTF-8。conn.php 的代码如下：

```
<?php
//连接数据库名为 oracle，用户名是 system，密码是 mrsoft
//如果 Oracle 数据库服务器与 PHP 程序处于同一机器中，在连接时数据库名称可以省略
$conn=oci_connect("system","mrsoft","oracle","utf8") or dir(oci_error());
?>
```

（2）创建 index.php 文件，包含数据库连接文件，执行查询语句，通过 oci_fetch_all()函数获取数据库中总的记录数，并且将查询结果存储到$results 数组中，最后通过 foreach 语句和 for 语句循环输出数组中的数据，其关键代码如下：

```
<?php
include("conn/conn.php");                                    //包含数据库连接文件
$stmt = oci_parse($conn, "select * from tb_user");           //预置 Oracle 查询语句
oci_execute($stmt);                                          //执行查询语句
$nrows = oci_fetch_all($stmt, $results);                     //获取查询结果的记录数
if ($nrows > 0) {                                            //判断查询结果
?>
<tr>
<?php
foreach ($results as $key => $val) {                         //循环输出数据库中字段的标题
?>
        <td height="30" align="center" valign="middle" bgcolor="#FFFFFF"><?php echo $key;?></td>
<?php
}
?>
</tr>
<?php
for ($i = 0; $i < $nrows; $i++) {                            //根据记录总数执行 for 循环
?>
<tr>
<?php
        foreach ($results as $data) {                        //输出数据库中的数据
?>
        <td align="center" valign="middle" bgcolor="#FFFFFF"><?php echo $data[$i];?></td>
<?php
        }
?>
```

```
</tr>
<?php
}
?>
<tr>
        <td height="35" colspan="5" align="right" bgcolor="#FFFFFF"><span class="STYLE1"><?php echo "总共有$nrows 记录!";?> </td>
</tr>
<?php
}else{
?>
<tr><td colspan="5" align="center" bgcolor="#FFFFFF">无数据</td>
</tr>
<?php
}
oci_close($conn);
?>
```

秘笈心法

心法领悟 525：oci_fetch_all()函数的应用总结。

应用 oci_fetch_all()函数获取数据库中总的记录时，在其返回的数组中不但包括数据表中的数据而且还包括数据表字段的名称。因此，在应用 oci_fetch_all()函数循环输出数据表中的数据时，不必单独定义数据表字段的名称表格，可以直接从数组中获取，同时也不必再应用其他函数对数据表中的总记录数进行统计。

实例 526 动态获取数据表中的字段和数据 高级

光盘位置：光盘\MR\12\526 趣味指数：★★★★

实例说明

如果采用固定字段名称的方法输出数据表中的数据，那么一旦数据表中的字段发生变化，再通过原来的表格输出数据表中的数据时，其运行效果将如图 12.7 所示，页面输出内容是不完整的。

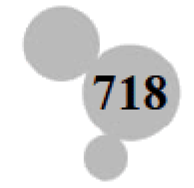


图 12.7　采用固定字段名称的方法输出数据表中的数据

如果采用动态读取数据表中的字段和数据的方法就可以避免图 12.7 所示的问题，无论字段发生怎样的变化，在页面中输出的数据都是完整的，不必对页面的设计进行重新调整，其运行效果如图 12.8 所示。



图 12.8　统计数据库中数据的记录总数



关键技术

在本实例中，首先应用 oci_num_fields()函数获取数据表中数据的列数，其语法如下：

```
int oci_num_fields( resource statement )
```

该函数返回参数 statement 中列的数目，其中参数 statement 为 oci_parse()函数返回的标识符。

然后应用 oci_field_name()函数动态获取数据表中的字段信息，其语法如下：

```
string oci_field_name( resource statement, int field )
```

该函数返回与字段数字索引（从 1 开始）相对应的字段名。

最后应用 oci_fetch_row()函数完成数据库中数据的循环输出，以数字作为数组的索引，其语法如下：

```
array oci_fetch_row( resource statement )
```

该函数从和指定的结果标识关联的结果集中获取一行数据并作为数组返回，如果没有更多行则返回 FALSE。每个结果的列存储在一个数组的单元中，数字索引从 0 开始。

通过本实例中介绍的函数循环输出数据库中的数据，可以避免因数据表增加字段或者减少字段而要重新设计页面的麻烦。

设计过程

（1）新建 conn 文件夹，编写 conn.php 文件，定义 Oracle 数据库的连接方法，设置数据库的编码格式为 UTF-8。

（2）新建 index.php 文件，首先包含数据库连接文件，然后应用 oci_num_fields()和 oci_field_name()函数以及 for 语句完成数据表中字段名称的输出。最后应用 oci_fetch_row()函数和 while 语句，根据数据表中字段的列数完成数据的循环输出，其关键代码如下：

```
<?php
include("conn/conn.php");                              //包含数据库连接文件
$stmt = oci_parse($conn, "select * from tb_user");     //定义查询语句
oci_execute($stmt);                                    //预置 Oracle 语句
$ncols = oci_num_fields($stmt);                        //获取数据表字段的列数
for ($i = 1; $i <= $ncols; $i++) {                     //执行 for 循环
 $field_name    = oci_field_name($stmt, $i);           //获取字段的名称
    echo "<td bgcolor='#FFFFFF' align='center' >$field_name</td>";
}
oci_free_statement($stmt);                             //释放关联于 Oracle 语句或游标的所有资源
?>
          </tr>
<?php
$query="select * from tb_user ";                       //定义 SQL 语句
$result=oci_parse($conn,$query);                       //预置 Oracle 语句
$r = oci_execute($result, OCI_DEFAULT);                //执行查询语句
while($row=oci_fetch_row($result)){                    //while 循环输出查询结果数组中的数据
?>
          <tr>

<?php
for ($i = 0; $i < $ncols; $i++) {                      //根据字段的列数输出数据
    echo "<td bgcolor='#FFFFFF' align='center' >$row[$i]</td>";
}

?>
          </tr>
<?php
}
oci_close($conn);                                      //关闭数据库
?>
```

秘笈心法

心法领悟 526：盘点本实例应用的函数。

oci_field_name()函数返回与字段数字索引（从 1 开始）相对应的字段名，即获取第一个字段的名称时，应

```
            $column_name = oci_field_name($stmt, $i);                    //获取字段名称
            $column_precision = oci_field_precision($stmt, $i);          //获取字段的精度
            $column_scale =oci_field_scale($stmt, $i);                   //获取字段的范围
            $column_type = oci_field_type($stmt, $i);            //字段的数据类型
            $column_size = oci_field_size($stmt, $i);            //获取字段的大小
            $column_type_raw = oci_field_type_raw($stmt, $i);    //获取字段的原始 Oracle 数据类型
        ?>
<tr>
    <td align="center" bgcolor="#FFFFFF"><?php echo $column_name ;?></td>
    <td align="center" bgcolor="#FFFFFF"><?php echo $column_precision ;?></td>
    <td align="center" bgcolor="#FFFFFF"><?php echo $column_scale ;?></td>
    <td align="center" bgcolor="#FFFFFF"><?php echo $column_type ;?></td>
    <td align="center" bgcolor="#FFFFFF"><?php echo $column_size ;?></td>
    <td align="center" bgcolor="#FFFFFF"><?php echo $column_type_raw ;?></td>
  </tr>
<?php
    }
    oci_free_statement($stmt);                                           //释放资源
    oci_close($conn);                                                    //关闭数据库
?>
</table>
```

秘笈心法

心法领悟 527：获取数据表中字段信息的函数。

获取数据表中的字段信息并不是一项常用的技术，它只会出现在一些特殊操作中，例如动态创建数据表、在网站的后台管理系统中查看数据表的结构等。其中只有 oci_num_fields ()和 oci_field_name()函数可能会经常使用，有关这些函数的详细讲解请参考 PHP 手册。

实例 528	修改 Oracle 数据库用户的密码 光盘位置：光盘\MR\12\528	高级 趣味指数：★★★☆

实例说明

在本实例中，介绍一种可以远程修改 Oracle 数据库用户密码的方法，其运行效果如图 12.10 所示。

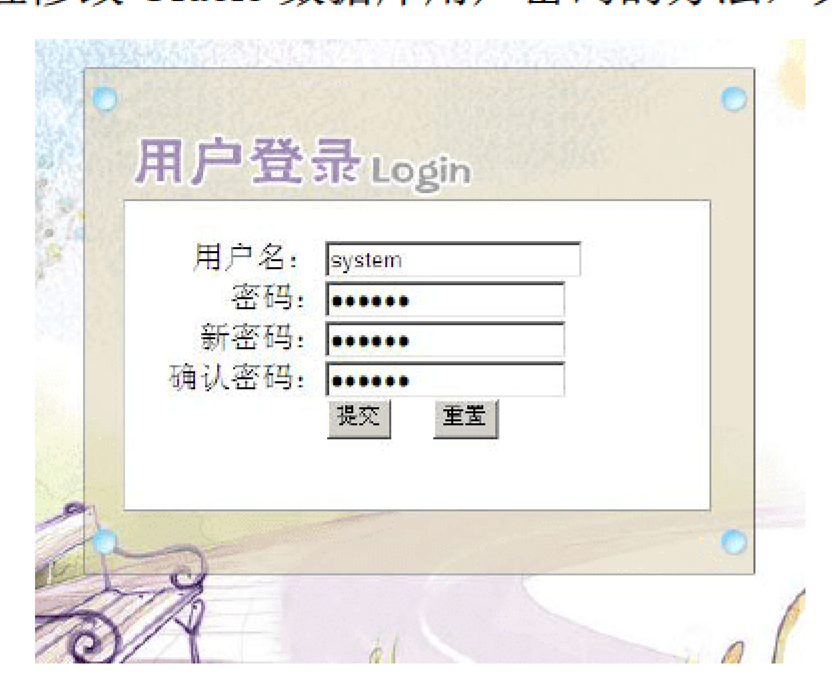


图 12.10　修改 Oracle 数据库用户的密码

关键技术

修改 Oracle 数据库用户的密码应用的是 oci_password_change()函数，其语法如下：

```
bool oci_password_change ( resource connection, string username, string old_password, string new_password )
```

oci_password_change()函数用于修改用户 username 的密码。如果成功返回 TRUE，失败则返回 FALSE。

oci_password_change()函数的参数说明如表 12.7 所示。

```
bool oci_fetch( resource statement )
```

在 PHP 5.0.0 之前的版本必须使用 ocifetch()替代本函数。ocifetch()函数名仍然可用，为向下兼容作为 oci_fetch()的别名，但不推荐使用。

设计过程

（1）新建 conn.php 文件，连接 Oracle 数据库，指定数据库编码格式为 UTF-8，其代码如下：

```
<?php
//连接数据库名为 oracle，用户名是 system，密码是 mrsoft
//如果 Oracle 数据库服务器与 PHP 程序处于同一机器中，在连接时数据库名称可以省略
$conn=oci_connect("system","mrsoft","oracle","utf8") or dir(oci_error());
?>
```

（2）新建 index.php 文件，通过 PHP 变量完成数据的循环输出。首先包含数据库连接文件，然后定义 SQL 语句，通过 oci_define_by_name()函数将 SQL 语句中查询出的字段名称与指定的变量进行绑定。最后执行查询操作，通过 while 语句循环输出查询结果，其关键代码如下：

```
<?php
   include("conn/conn.php");                                    //包含数据库连接文件
      $query="select name, position, age from tb_user ";        //定义 SQL 语句
      $result=oci_parse($conn,$query);                          //预置 Oracle 语句
      oci_define_by_name($result, "NAME", $name);               //将字段 name 与变量$name 绑定
      oci_define_by_name($result, "POSITION", $position);
      oci_define_by_name($result, "AGE", $age);
      oci_execute($result);                                     //执行查询操作
      while (oci_fetch($result)) {                              //循环输出查询结果
 ?>
  <tr>
    <td align="center" valign="middle"> </td>
      <td align="center" valign="middle"><span class="STYLE6"><?php echo $name;?></span></td>
<td align="center" valign="middle"><span class="STYLE6"><?php echo $position;?></span></td>
<td align="center" valign="middle"><span class="STYLE6"><?php echo $age;?></span></td>
  </tr>
<?php
}
     oci_free_statement($result);                               //释放空间
     oci_close($conn);                                          //关闭数据库
?>
```

秘笈心法

心法领悟 530：应用 oci_define_by_name()函数的注意事项。

（1）Oracle 使用大写字母的列名，而用户在 select 中可以用小写字母。

（2）如果需要定义一个抽象数据类型（LOB/ROWID/BFILE），必须先用 oci_new_descriptor()函数分配空间。

12.3　Oracle 数据库应用

前面只是讲解连接 Oracle 数据库的方法以及一些简单的操作，在本节中将 Oracle 数据库应用到实际的程序开发中。

实例 531	Oracle——管理员注册功能 光盘位置：光盘\MR\12\531	高级 趣味指数：★★★★☆

实例说明

本实例实现一个管理员的注册功能，应用 ADODB 连接、操作 Oracle 数据库，将注册信息添加到指定的

```
<tr>
        <td align="right">验证码：</td>
           <td align="left">
                <input name="check_code" type="text" class="input" id="check_code" size="10" />
                <input type="hidden" name="defValidatorCode" value="" />
<script language="javascript">
 var num1=Math.round(Math.random()*10000000);
      var num=num1.toString().substr(0,4);
      document.write("<img name=codeimg src='ValidatorCode.php?code="+num+"'>");
      form1.defValidatorCode.value=num;
 function reCode(){
         var num1=Math.round(Math.random()*10000000);
       var num=num1.toString().substr(0,4);
       document.codeimg.src="ValidatorCode.php?code="+num;
       form1.defValidatorCode.value=num;
 }
</script>
 <a href="javascript:reCode()" class="a1">看不清</a>
        </td>
 </tr>
</table>
</form>
```

（3）创建 index_ok.php 文件，将表单中提交的数据添加到指定的数据表中。在处理密码时应用 MD5 对密码进行加密，由于页面中的编码格式是 UTF-8，而 Oracle 数据库中使用的是 GB2312，为了避免数据输出时出现乱码，在执行添加操作之前，应用 iconv()函数将所有提交的数据转换为 GB2312 编码，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" ); //设置文件编码格式
include("conn/conn.php");
if($_POST['name']!="" && $_POST['pass']!=""){
$sql="insert into tb_admin(id,name,pass,dates)
values(tb_adminid.nextval,'".iconv("utf-8","gb2312",$_POST['name'])."','".md5(iconv("utf-8","gb2312",$_POST['pass']))."','".date("Y-m-d")."')";
$rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );
if($rst){
            echo "<script>alert('管理员添加成功！');window.location.href='index.php?conn_id=1';</script>";
}else{
            echo "<script>alert('管理员添加失败！');windwo.location.href='index.php'</script>";
}
}else{
      echo "<script>alert('请输入管理员信息！');windwo.location.href='index.php'</script>";
}
?>
```

（4）创建 ValidatorCode.php 文件，定义验证码的实现方法，详细内容请参考关键技术。

（5）创建 js 脚本文件夹，编写 check.js 脚本文件，定义 chkinput()方法对表单中的元素值进行验证。

秘笈心法

心法领悟 531：Oracle 数据库中 insert 添加语句的编写。

在创建数据表时，指定 ID 字段为表的主键，通过序列完成主键 ID 值的添加，在本实例中序列为 tb_adminid.nextval。

实例 532	Oracle——管理员登录 光盘位置：光盘\MR\12\532	高级 趣味指数：★★★✫

实例说明

下面讲解 Oracle 数据库中管理员登录功能的实现，将详细讲解如何在 Oracle 数据库中完成用户名和密码的验证操作，本实例的运行结果如图 12.14 所示。

秘笈心法

心法领悟 532：编码格式转换的必要性。

在创建的 Oracle 数据库中，数据的编码格式是 GB2312，而程序页面中的编码格式是 UTF-8，所以在执行 select 查询语句时，在 where 条件中，必须应用 iconv()函数对表单中提交的用户名和密码进行编码转换，使其与数据库中存储的数据编码统一，这样才可以进行比较。否则由于数据的编码格式不同，无论输入的用户名和密码是否正确都不能成功登录。唯一一种对数据编码格式没有限制的情况就是数据中不包含中文字符串。

实例 533	Oracle——管理员数据管理 光盘位置：光盘\MR\12\533	高级 趣味指数：★★★★☆

实例说明

本实例在实现管理员登录功能的基础上，增加对管理员数据的更新和删除功能。可以对管理员的名称、密码和注册时间进行更新操作，同时也可以删除指定的管理员数据，其运行结果如图 12.15 所示。



图 12.15　管理员数据的更新和删除

关键技术

这里仍应用 ADODB 连接、操作 Oracle 数据库。以数据表中数据的 ID 为条件，执行 update 更新和 delete 删除语句。

在本实例中，管理员登录成功后，应用 SESSION 会话机制将登录的管理员名称存储到 SESSION 变量中，通过这个 SESSION 来控制用户的权限，如果是管理员则可以执行管理员数据的更新和删除操作，否则将给出提示信息，并跳转到管理员登录页面，其关键代码如下：

```
<?php
session_start();                                    //初始化 SESSION 变量
header ( "Content-type: text/html; charset=utf-8" );    //设置文件编码格式
if($_SESSION['name']!=""){                          //判断管理员的名称是否为空
include("conn/conn.php");                           //包含数据库连接文件
if($_POST['Submit2']=="删除"){
    $sql="delete from tb_admin where id='".$_POST['conn_id']."'";
    $rst = $conn->execute ( $sql ) or die ( 'error：  ' . $conn->errorMsg () );
    if($rst){
        echo "<script>alert('管理员删除成功！');window.location.href='main.php';</script>";
    }else{
```

SESSION 变量判断当前用户是否具有操作权限，如果有权限，则可以根据按钮的值执行更新或者删除操作；否则将给出提示信息，并跳转到管理员登录页面，其代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=utf-8" );              //设置文件编码格式
if($_SESSION['name']!=""){
include("conn/conn.php");
if($_POST['Submit2']=="删除"){
$sql="delete from tb_admin where id='".$_POST['conn_id']."'";
$rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );
if($rst){
        echo "<script>alert('管理员删除成功！');window.location.href='main.php';</script>";
}else{
        echo "<script>alert('管理员删除失败！');windwo.location.href='main.php'</script>";
}
}
if($_POST['Submit']=="更新"){
$sql="update tb_admin set
name='".iconv("utf-8","gb2312",$_POST['name'])."',pass='".md5(iconv("utf-8","gb2312",$_POST['pass']))."',dates='".$_POST['dates']."' where
id='".$_POST['conn_id']."'";
$rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );
if($rst){
        echo "<script>alert('管理员更新成功！');window.location.href='main.php';</script>";
}else{
        echo "<script>alert('管理员更新失败！');windwo.location.href='main.php'</script>";
}
}
}else{
echo "<script>alert('您不具备管理员权限！');windwo.location.href='index.php'</script>";
}
?>
```

秘笈心法

心法领悟 533：SESSION 变量的应用。

通过 SESSION 变量的值来判断当前用户的权限是非常实用的方法，其原理是：在会员或者管理员登录成功后，将登录的名称或者一个指定的值赋给 SESSION 变量，由于 SESSION 变量可以实现在不同页面之间的传递，所以可以在具有访问权限限制的页面中判断这个 SESSION 变量的值，如果变量值为真，则说明这个用户具有访问权限，否则说明当前用户不具备访问权限。

实例 534	Smarty+ADODB 完成 Oracle 数据的分页显示 光盘位置：光盘\MR\12\534	高级 趣味指数：★★★☆

实例说明

在本实例中，同样应用 ADODB 连接、操作 Oracle 数据库，其不同之处是将连接、操作数据库的方法封装到类中，同时将数据库中数据的分页方法也封装到类中，还有通过 Smarty 模板完成页面的动静分离。本实例完成数据库中员工信息的分页输出，其运行结果如图 12.16 所示。

ID	名称	职务	年龄	入职时间
5	欧阳	部门经理	28	2010-08-20
6	东方	设计师	28	2010-08-20
8	明日科技	总经理	28	2008-08-10

共有员工 8 名 每页显示 3 名 第 2 页/共 3 页　首页 上一页 下一页 尾页

图 12.16　Smarty+ADODB 完成 Oracle 数据的分页显示

关键技术

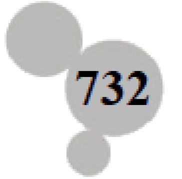
本实例的关键是应用面向对象技术，将 Oracle 数据库的连接、操作以及分页方法都封装到类中，同时还包括 Smarty 模板的配置方法。应用类的实例化返回的对象调用类中方法，完成与 Oracle 数据库的连接、操作和分页，最终通过 Smarty 模板完成页面的动静分离。

（1）Oracle 数据库连接类 ConnDB，不但可以连接 Oracle 数据库，还可以连接 MySQL、SQL Server 和 Access 数据库。该类存储于 system\system.smarty.inc.php 文件中，其内容如下：

```
class ConnDB{
	var $dbtype;
	var $host;
	var $user;
	var $pwd;
	var $dbname;
	var $debug;
	var $conn;
	function ConnDB($dbtype,$host,$user,$pwd,$dbname,$debug=false){			//构造方法，为成员变量赋值
		$this->dbtype=$dbtype;
		$this->host=$host;
		$this->user=$user;
		$this->pwd=$pwd;
		$this->dbname=$dbname;
		$this->debug=$debug;
	}
	function GetConnId(){								//实现与不同数据库的连接并返回连接对象
		require("../adodb5/adodb.inc.php");					//调用 ADODB 类库文件
		if($this->dbtype=="oci8" || $this->dbtype=="mysql" || $this->dbtype=="mssql"){	//判断成员变量传递的数据库类型
			if($this->dbtype=="mysql"){					//判断如果是 MySQL 数据库
				$this->conn=NewADOConnection("mysql");			//执行与 MySQL 数据库的连接
			}else if($this->dbtype=="mssql"){
				$this->conn=NewADOConnection("mssql");
			$this->conn->Connect($this->host,$this->user,$this->pwd,$this->dbname);	//数据库连接的用户、密码
			}else{
				$this->conn = ADONewConnection("oci8");			//建立一个 Oracle 连接
				$this->conn->Connect($this->dbname,$this->user,$this->pwd);
			}
		}elseif($this->dbtype=="access"){					//判断如果使用的是 Access 数据库
			$this->conn=NewADOConnection("access");
			$this->conn->Connect("Driver={Microsoft Access Driver
(*.mdb)};Dbq=".$this->dbname.";Uid=".$this->user.";Pwd=".$this->pwd.";");		//执行连接 Access 数据库
		}
		$this->conn->Execute("set names utf8");				//设置数据库的编码格式
		if($this->dbtype=="mysql")
			$this->conn->debug=$this->debug;
		return $this->conn;							//返回连接对象
	}
	function CloseConnId(){								//定义关闭数据库的方法
		$this->conn->Disconnect();						//执行关闭的操作
	}
}
```

（2）数据库操作类 AdminDB，执行对 Oracle 数据库的操作语句，如果是 select 语句，成功则返回结果集数组，否则返回 FALSE；如果是 update、insert 或者 delete 语句，成功则返回 TRUE，否则返回 FALSE。该类存储于 system\system.smarty.inc.php 文件中，其内容如下：

```
class AdminDB{
	function ExecSQL($sqlstr,$conn){					//定义方法，参数为 SQL 语句和连接数据库返回的对象
		$sqltype=strtolower(substr(trim($sqlstr),0,6));		//截取 SQL 中的前 6 个字符串，并转换成小写
		$rs=$conn->Execute($sqlstr);					//执行 SQL 语句
		if($sqltype=="select"){						//判断如果 SQL 语句的类型为 select
			$array=$rs->GetRows();					//执行该语句，获取查询结果
			if(count($array)==0 || $rs==false)			//判断语句是否执行成功
				return false;					//如果查询结果为 0，或者执行失败，则返回 FALSE
			else
```



```
                <td align="center" bgcolor="#FFFFFF">{Util text=$arr[arrid].4}</td>
</tr>
        {/section}
{/if}
{$showpage}
```

秘笈心法

心法领悟 534：ADODB 类库和 Smarty 模板。

有关 ADODB 类库的具体应用将在第 13 章中进行详细讲解，而 Smarty 模板的应用可以参考本书第 14 章。在本实例中，封装数据库连接、操作类、分页类和 Smarty 模板的配置类是一个非常不错的方法，推荐读者使用，并且可以对它进行不断地完善。例如，扩展其支持更多的数据库、更改分页类的方法以及更改 Smarty 模板中配置文件的存储路径等。

实例 535	Oracle——存储过程更新员工信息 光盘位置：光盘\MR\12\535	高级 趣味指数：★★★★☆

实例说明

在本实例中，应用 Oracle 数据库的存储过程技术，实现对员工信息表中数据的更新操作，其运行结果如图 12.17 所示。



图 12.17　存储过程更新员工信息

关键技术

存储过程是为执行一定的任务而组合在以前的 SQL 和/或 PL/SQL 语句集，通常由以下几部分组成：声明部分、执行部分和异常处理部分，其中执行语句是必需的，而其他两部分则是可选的。

创建存储过程的前提条件是用户具有创建过程的权限，这个权限可以由 DBA 赋予用户，其语法如下：

```
GRANT CREATE ANY PROCEDURE TO UserName;
```

取消权限的语法如下：

```
REVOKE CREATE ANY PROCEDURE FROM UserName;
```

Oracle 数据库中创建存储过程的语法如下：

```
CREATE[OR REPLACE] PROCEDURE procedure_name(
argument1[{in|out|in out}] type,
argument2[{in|out|in out}] type,
…
Argument n[{in|out|in out}] type
)
{is | as}
<!--类型变量的说明-->
BEGIN
<!--执行部分-->
```



```
[EXCEPTION]
<!--可选的异常处理部分-->
END[procedure_name];
```

参数说明：

REPLACE：关键字，为可选值，多数情况下会使用。其作用为创建存储过程时，如果同名过程已经存在则删除同名过程，然后进行创建操作。有一点需要注意，同一用户下的过程是唯一标识的，如果创建时不使用 REPLACE 关键字，当出现同名过程时，会给出 ORA-00955 错误，表示名称已经被现有对象使用。

procedure_name：创建过程的名称，它在数据库中同一用户下是唯一标识。

argument：过程的参数名称，为可选参数。

in、out、in out：参数的模式，默认的模式为 in。in 模式为输入参数模式，具备只读属性，不可以在执行过程中赋值；out 模式为输出参数模式；in out 为输入/输出参数模式。

type：指定参数的数据类型。其对形参的定义只可以指定类型，不可以限定精度。

is|as：该关键字到 BEGIN 关键字之间为过程的声明部分。声明包括类型、游标、常量、变量、异常和嵌套子程序的声明，此处不需 DECLARE 关键字，而且此声明部分中声明的都是本地和局部声明，即过程结束时声明终止。

BEGIN 关键字和 END 关键字之间为过程的执行部分，如果过程中存在异常处理，则 BEGIN 和 EXCEPTION 关键字之间为过程的执行部分，EXCEPTION 和 END 关键字之间为异常处理部分。

在本实例中创建的存储过程如下：

```
CREATE OR REPLACE PROCEDURE PROC_USER (
   p_id IN integer,
   p_position IN VARCHAR
) AS
BEGIN
   update tb_user set position=p_position where id=p_id;
  commit;
END PROC_USER;
```

设计过程

（1）创建 conn 文件，编写 conn.php 文件，完成与 Oracle 数据库的连接。

（2）创建 index.php 文件，包含数据库连接文件，应用 Oracle 函数读取 Oracle 数据库中的数据，并通过 while 语句完成数据的循环输出。创建 form 表单，完成员工更新信息的提交，将更新数据提交到 index_ok.php 文件中，其关键代码如下：

```
<?php
include("conn/conn.php");
$query="select * from tb_user ";
$result=oci_parse($conn,$query);
$r = oci_execute($result, OCI_DEFAULT);
while($row=oci_fetch_array($result,OCI_RETURN_NULLS)){
?>
<form id="form<?php echo $row[0];?>" name="form<?php echo $row[0];?>" method="post" action="update_ok.php">
<tr>
<td align="center" bgcolor="#FFFFFF"><?php echo $row[0];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $row[1];?></td>
      <td align="center" bgcolor="#FFFFFF">
         <input name="position" type="text" value="<?php echo $row[2];?>" size="15" />
            <input type="hidden" name="conn_id" value="<?php echo $row[0];?>" /></td>
         <td align="center" bgcolor="#FFFFFF"><?php echo $row[3];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $row[4];?></td>
      <td align="center" bgcolor="#FFFFFF"><input type="submit" name="Submit" value="更新" /></td>
</tr>
</form>
<?php
}
oci_close($conn);
?>
```

（3）创建 index_ok.php 文件，获取表单中提交的数据，调用存储过程完成员工信息的更新操作，其关键代

码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );                    //设置文件编码格式
include("conn/conn.php");
$sql = "BEGIN PROC_USER(:id, :position); END;";                        //定义 SQL 语句
$stmt = OCIParse($conn, $sql);                                          //执行 SQL 语句
//执行绑定
OCIBindByName($stmt, ":id", $_POST['conn_id'], 4);                      //参数说明：绑定 PHP 变量 conn_id 到位置:id，并设定绑定长度为 4 位
OCIBindByName($stmt, ":position", $_POST['position'], 50);
if(OCIExecute($stmt)){                                                  //调用并执行存储过程
 echo "<script>alert('员工信息更新成功!');window.location.href='index.php';</script>";
}else{
	echo "<script>alert('员工信息更新失败!');window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 535：在 PHP 动态页中调用存储过程。

首先定义 SQL 语句，执行存储过程中的更新语句。然后通过 ociparse()函数配置 Oracle 语句预备执行。接着通过 OCIBindByName()函数绑定一个 PHP 变量到一个 Oracle 标识符。最后通过 OCIExecute()函数执行 Oracle 语句，完成员工信息的更新操作。

12.4　Oracle 数据的导入和导出

本节将介绍 Oracle 数据库中数据的导入、导出方法。

实例 536	Oracle 数据库中的数据导出至文本文件 光盘位置：光盘\MR\12\536	高级 趣味指数：★★★☆

实例说明

本实例将 Oracle 数据库中的数据表导出到本地磁盘 C 盘下的 mrsoft 文件夹中，并将导出的文本文件命名为 oracle.dmp。导出的结果如图 12.18 所示。

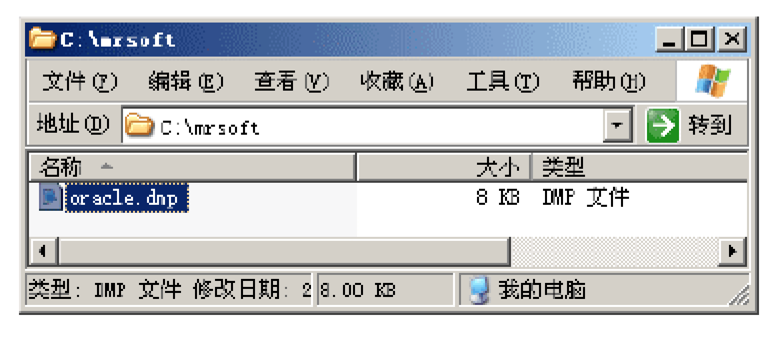


图 12.18　导出的文本文件

关键技术

本实例应用 Oracle 数据库中数据库逻辑备份工具 Export。利用 Export 工具可以在数据库打开状态下备份数据库。Export 把数据库中的对象导出到一个二进制文件中。该工具也是数据库间进行迁移的一个工具。

Export 有 3 种方法：行命令法、交互式方法和参数文件方法。下面主要介绍本实例中使用的参数文件方法。

将 EXP 命令的各个参数放在一个参数文件中，参数文件的命令格式如下：

```
EXP [用户名| 口令] PARFILE = 参数文件路径及名称
```

以下是一个参数文件的示例，代码如下：

```
FULL = Y
FILE = expfile.dmp
GRANTS= Y
INDEXES= Y
```

其中，FULL 设置为 Y 时，会导出除 SYS 方案对象外的所有的方案对象，如果设置为 FULL，要求用户有 EXP_FULL_DATABASE 权限；FILE 参数用来设置导出文件的目录和文件名；GRANTS 参数默认值为 Y，用来设置是否导出权限信息；INDEXES 参数默认值为 Y，用来设置是否导出表和聚集的索引。



关键技术

（1）导入用户自身所拥有的表，其命令如下：

```
imp scott/tiger tables=(name,emp) file=export.dmp
```

（2）当用户要导出其他方案的表时，必须具备 DBA 角色，或者被赋予 EXP_FULL_DATABASE 角色，其命令如下：

```
imp system/manger tables=(scott.name, scott.emp) file=export.dmp
```

（3）当执行导入操作时，用户可以只导入表结构而不导入数据，其命令如下：

```
imp scott/tiger tables=(name, emp) file=export.dmp rows=n
```

（4）当执行导入操作时，如果对象已经存在就可以只导入数据，其命令如下：

```
imp system/manager tables=(scott.name,scott.emp) file=export.dmp ignore=y
```

设计过程

（1）选择“开始”/“程序”/“附件”/“命令提示符”命令，进入 DOS 命令环境。

（2）在 DOS 提示符下输入“IMP 用户名|口令命令”，将文本文件中的数据导入到 Oracle 数据库中，代码如下：

```
imp system/mrsoft tables=( tb_user,tb_admin) file= C:\mrsoft\oracle.dmp
```

（3）按 Enter 键，执行本实例，其运行结果如图 12.20 所示。

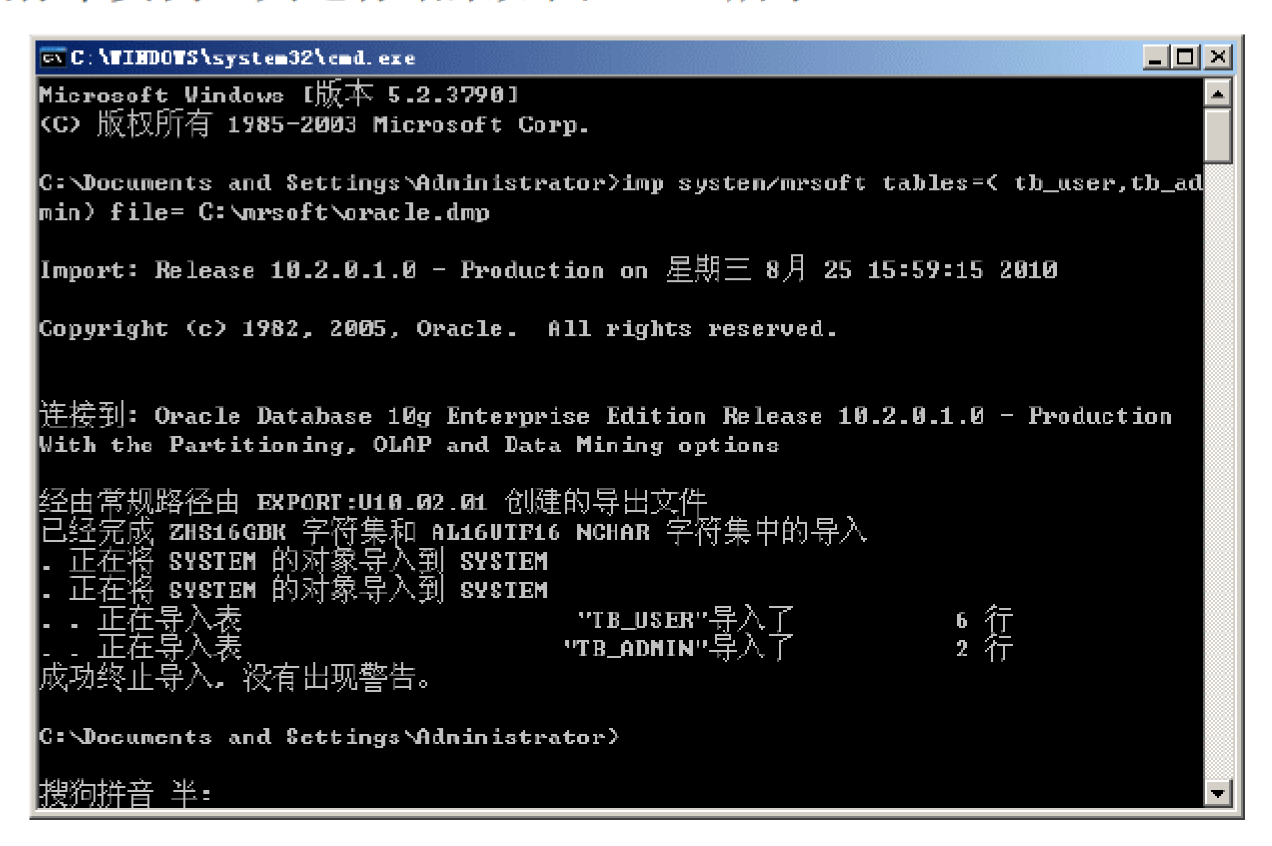


图 12.20　文本文件中的数据导入至 Oracle

秘笈心法

心法领悟 537：导入表空间和导入整个数据库。

通过 import 工具不但可以导入表，还可以将数据导入至不同的账户，导入模式、表空间和整个数据库。

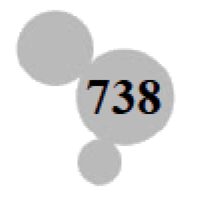
导入整个数据库是应用 import 工具将文件中的对象及数据导入到数据库中。

（1）执行导入时，需要导入数据库损坏前最新的信息，其命令如下：

```
imp system/manager full=y file=export.dmp
```

（2）执行导入时，需要导入最新的完全导出文件信息，其命令如下：

```
imp system/manager full=y file=export.dmp
```





新技术篇

第 13 章

ADODB 类库

- 封装 ADODB 操作类
- ADODB 的应用
- ADODB 中的错误处理
- ADODB 操作数据库

13.1　封装 ADODB 操作类

ADODB 是 Active Data Objects Data Base 的简称，是一种 PHP 存取数据库的中间函式组件。本节将介绍如何对 ADODB 操作数据库的方法进行封装，从而使通过 ADODB 操作数据库的方法更加简单。

说明：因为 ADODB 属于第三方组件，所以在本书的光盘中没有提供 ADODB 类库文件，需要读者自行下载，并复制到本章的 13 文件夹下，将其命名为 adodb5，然后才可以运行本章的实例。

实例 538	ADODB 连接数据库类 光盘位置：光盘\MR\13\538	高级 趣味指数：★★★★☆

实例说明

连接数据库类的封装，意味着通过 ADODB 连接不同的数据库的方法更加简单、更容易更新和修改。只要向类中传递对应的参数值，即可完成与不同数据库服务器的连接。在本实例中，以 MySQL 数据库服务器为例，实现与 db_database13 数据库的连接，连接成功后弹出提示信息，其运行结果如图 13.1 所示。

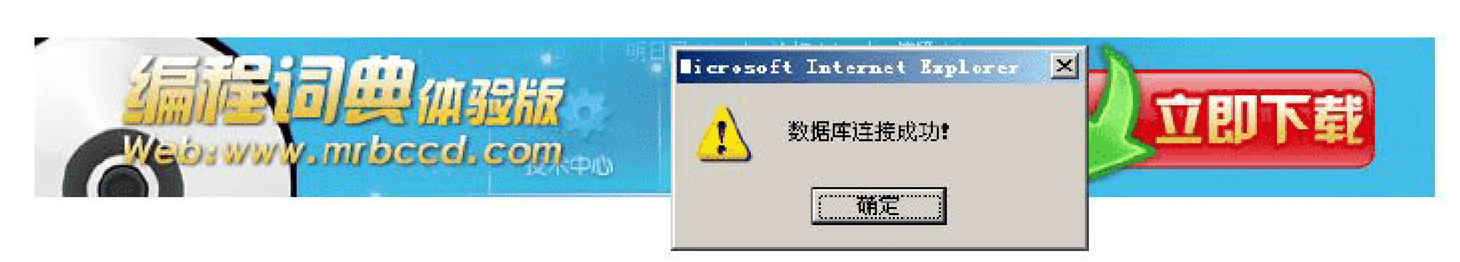


图 13.1　ADODB 连接数据库类

关键技术

ADODB 连接数据库系统使用 ADONewConnection()函数；如果选择持久化连接则使用 PConnect()函数，如果选择非持久化连接则使用 Connect()函数。

（1）ADONewConnection()函数连接数据库系统，其语法如下：

```
ADONewConnection($databaseType)
```

参数$databaseType 表示要连接的数据库系统的名称，如 mysql、mssql 等。

（2）PConnect()函数实现与数据库的持久化连接，其语法如下：

```
PConnect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

$database：使用到的数据库。

（3）Connect()函数实现与数据库的非持久化连接，其语法如下：

```
Connect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

$database：使用到的数据库。

```
}else{
echo "<script>alert('数据库连接失败！');window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 538：通过 ADODB 连接数据库服务器必须注意的问题。

定义连接数据库类，再定义连接方法 GetConnId()时，必须先通过 require 语句包含 ADODB 类库文件 adodb.inc.php，否则其他一切操作都无从谈起。

建议在程序的开发阶段运用连接数据库操作类时，将参数$debug 的值设置为 TRUE，这样可以返回执行的 SQL 语句，便于获取 SQL 语句在执行过程中出现的错误。如果是在生产环境中运用连接数据库类，那么就要将$debug 的值设置为 FALSE。

实例 539　**ADODB 操作数据库类**　高级

光盘位置：光盘\MR\13\539　趣味指数：★★★★☆

实例说明

既然可以将 ADODB 连接数据库的方法封装到类中，那么同样也可以将 ADODB 操作数据库的方法封装到类中。本实例介绍如何将 ADODB 操作数据库的方法封装到类中，并且通过该方法读取 MySQL 数据库服务器的 db_database13 数据库中 tb_bccd 数据表中的数据，其运行结果如图 13.2 所示。

ID	名称	版本	时间
1	VC编程词典	标准版	2010-07-29
2	VB编程词典	标准版	2010-07-29
3	Java编程词典	标准版	2010-07-29
4	C#编程词典	标准版	2010-07-29
5	PHP编程词典	标准版	2010-07-29
6	C++编程词典	标准版	2010-07-29
7	Java2编程词典	标准版	2010-07-29
8	C编程词典	标准版	2010-07-29
9	ASP编程词典	标准版	2010-07-29
10	办公自动化编程词典	标准版	2010-07-29
11	硬件维护编程词典	标准版	2010-07-29
12	电脑基础编程词典	标准版	2010-07-29

图 13.2　ADODB 操作数据库类

关键技术

ADODB 操作数据库类为 AdminDB，在操作数据库的 ExecSQL()方法中，关键是以下几个函数的运用。

（1）strtolower()函数将字符串转换为小写字母，其语法如下：

```
string strtolower(string str)
```

运用本函数的目的是将 SQL 语句中的字符串都转换为小写。

（2）substr()函数从指定的字符串中按照指定的位置截取一定长度的字符，其语法如下：

```
string substr(string str,int start,int length)
```

参数说明：

str：指定截取的字符串。

start：指定开始截取字符串的位置，如果为负数，则从字符串的末尾开始截取。

length：可选参数，指定截取字符串的长度。如果为负数，则表示取倒数第 length 个字符。

注意：本函数中参数 start 的指定位置是从 0 开始计算的，即字符串中的第一个字符表示为 0。

运用 substr()函数对 SQL 语句中前 6 个字符串进行截取，其目的是获取 SQL 语句的类型（insert、select 或者 update），判断其执行的是什么操作。

（3）trim()函数删除字符串中首尾的空白或者其他字符。

（4）execute()函数为 ADODB 中执行 SQL 语句的方法并返回一个结果集（ADORecordSet 对象），失败则返回 FALSE，其语法如下：

```
execute($sql[,$inputarr=false])
```

参数说明：

$sql：指定要执行的 SQL 语句。

$inputarr：设置传入的结合变量，如果没有设置$inputarr，则$sql 为普通的 SQL 语句，execute()函数的格式如下：

```
$connect->execute('select * from tb_user where id = 1')
```

如果设置了$inputarr，则 execute()函数的格式如下：

```
$connect -> execute('select * from tb_user where id = ?',array($vl))
```

（5）GetRows()函数是 GetArray()的同义函数，是为了与 Microsoft ADO 相容定义的，其语法如下：

```
GetRows ([$number_of_rows])
```

返回从当前指针指向的记录开始，到$number_of_rows – 1 行的全部记录的数组。参数$number_of_rows 表示指定的记录行，如果没有给出，则一直到 EOF 才停止。

设计过程

（1）创建 found.database.php 文件，定义数据库操作类 AdminDB，定义 ExecSQL()方法，根据传递的 SQL 语句执行对数据库的操作。在 ExecSQL()方法中，应用 substr()函数获取参数传递的 SQL 语句的类型，然后根据 SQL 语句的类型进行判断，如果是 select 查询语句，则执行 GetRow()方法，获取查询结果；如果是 update、insert 或者 delete 语句，则直接返回 TRUE 或者 FALSE，其代码如下：

```
class AdminDB{
function ExecSQL($sqlstr,$conn){                                  //定义方法，参数为 SQL 语句和连接数据库返回的对象
        $sqltype=strtolower(substr(trim($sqlstr),0,6));            //截取 SQL 中的前 6 个字符串，并转换成小写
        $rs=$conn->Execute($sqlstr);                               //执行 SQL 语句
        if($sqltype=="select"){                                    //判断如果 SQL 语句的类型为 select
                $array=$rs->GetRows();                             //执行该语句，获取查询结果
                if(count($array)==0 || $rs==false)                 //判断语句是否执行成功
                        return false;                              //如果查询结果为 0 或者执行失败，则返回 FALSE
                else
                        return $array;                             //否则返回查询结果的数组
        }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
                //判断如果 SQL 语句类型不为 select、则执行如下操作
                if($rs)
                    return true;                                   //执行成功返回 TRUE
                else
                    return false;                                  //否则返回 FALSE
        }
}
}
```

（2）创建 connect.database.php 文件，实例化数据库连接类 ConnDB 和数据库操作类 AdminDB，其关键代码如下：

```
<?php
require("found.database.php");                                     //包含类文件
$connobj=new ConnDB("mysql","localhost","root","111","db_database13",false);   //实例化数据库连接类
$conn=$connobj->GetConnId();                                       //返回连接标识
$admindb=new AdminDB();                                            //实例化数据库操作类
?>
```



$nrows：每页显示的记录数。

$page：保存当前页数，默认为 1。

（2）AbsolutePage()函数返回当前页的页数，其语法如下：

```
AbsolutePage($page=-1)
```

该函数要与 PageExecute()函数配合使用。

（3）AtFirstPage()函数为如果当前页是第一页，则返回 TRUE，其语法如下：

```
AtFirstPage($status='')
```

该函数要与 PageExecute()函数配合使用，判断当前是否为第一页。

（4）AtLastPage()函数判断当前页是否为最后一页，其语法如下：

```
AtLastPage($status='');
```

该函数同样要与 PageExecute()函数配合使用，判断当前页是否为最后一页，如果是则返回 TRUE。

本实例中整合了上述几个 ADODB 类库中的函数，封装了一个 ADODB 分页类。

设计过程

（1）创建 found.database.php 文件，定义 SepPage 分页类，定义 ShowData()方法，读取数据库中存储的数据，并且将查询结果返回到数组中。定义 ShowPage()方法，创建分页超链接，完成从数据库中读取数据的分页输出。SepPage 分页类的关键代码如下：

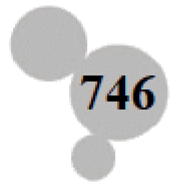
```
class SepPage{
 var $rs;
 var $pagesize;
 var $nowpage;
 var $array;
 var $conn;
 var $sqlstr;
 function ShowData($sqlstr,$conn,$pagesize,$nowpage){                //定义方法
        if(!isset($nowpage) || $nowpage=="")                         //判断变量值是否为空
               $this->nowpage=1;                                     //定义每页起始页
        else
               $this->nowpage=$nowpage;
        $this->pagesize=$pagesize;                                   //定义每页输出的记录数
        $this->conn=$conn;                                           //连接数据库返回的标识
        $this->sqlstr=$sqlstr;                                       //执行的查询语句
        $this->rs=$this->conn->PageExecute($this->sqlstr,$this->pagesize,$this->nowpage);
        @$this->array=$this->rs->GetRows();                          //获取记录数
               if(count($this->array)==0 || $this->rs==false)
                      return false;
               else
                      return $this->array;
 }
//省略了 ShowPage()方法中的内容
}
```

（2）创建 connect.database.php 文件，实例化数据库连接类 ConnDB、数据库操作类 AdminDB 和分页类 SepPage，其关键代码如下：

```
<?php
require("found.database.php");                                       //包含类文件
$connobj=new ConnDB("mysql","localhost","root","111","db_database13",false);  //实例化数据库连接类
$conn=$connobj->GetConnId();
$admindb=new AdminDB();                                              //实例化数据库操作类
$seppage=new SepPage();                                              //实例化分页类
?>
```

（3）创建 index.php 文件，包含 connect.database.php 文件，调用分页类中的 ShowData()方法，完成数据的分页输出；调用分页类中的 ShowPage()方法输出分页的超链接，其关键代码如下：

```
<?php
include_once 'conn/connect.database.php';                            //调用类中方法
$array=$seppage->ShowData("select * from tb_user",$conn,10,$_GET["page"]);  //分页读取数据库中数据
for($i=0;$i<count($array);$i++){                                     //循环输出数据库中数据
?>
```



ADODB 类库中提供的 ADODB_pager 类实现分页功能，分页输出编程词典的注册用户，其运行结果如图 13.5 所示。



图 13.5　ADODB_pager 分页类

关键技术

ADODB_Pager()函数是 ADODB_Pager 类的构造函数，通过类中的 render()函数可以实现分页功能，其语法如下：

```
ADODB_Pager($conn, $sql, $id = 'adodb', $showPageLinks = false)
```

参数$conn 是数据库连接对象；参数$sql 是执行的 SQL 语句；参数$id 为每个分页的 id 号；参数$showPageLinks 设置是否显示各个页的链接，默认为 FALSE。

提示：应用 ADODB_pager()函数实现分页功能的前提条件是必须载入 adodb-pager.ini.php 文件。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，载入 adodb.ini.php 文件。实现与 MySQL 数据库服务器中 db_database13 数据库的连接，并设置数据库的编码格式为 UTF-8，其代码如下：

```
<?php
include_once ('../adodb5/adodb.inc.php');                    //载入（include）adodb.inc.php 文件
$conn = ADONewConnection('mysql');                           //建立连接
$conn -> PConnect('localhost','root','111','db_database13'); //连接数据库
$conn -> execute('set names utf8');                          //设置编码格式

?>
```

（2）创建 index.php 文件，完成数据的分页输出。首先包含数据库连接文件，然后载入 adodb-pager.ini.php 文件，最后实例化 ADODB_Pager 类，调用 Render()方法，设置每页显示 10 条记录，其关键代码如下：

```
<?php
include 'conn/conn.php';                                     //载入数据库连接文件
include_once ' adodb-pager.inc.php';                         //载入 adodb-pager.ini.php 文件
/*   声明对象   */
$pager = new ADODB_Pager($conn,"select id,user as '用户名', pass as '密码',dates as '注册时间' from tb_user");
/*   调用 Render()函数确定每页显示多少记录   */
$pager -> Render(10);
?>
```

秘笈心法

心法领悟 542：修改 ADODB_Pager 类中定义的分页方法的样式。

修改 ADODB_Pager 类中定义的分页方法的样式，即对载入的 adodb-pager.ini.php 文件进行修改。可以修改的内容包括超链接、表格背景、宽度、边框宽度和颜色等。在本实例中，将超链接修改为中文并且重新定义了表格的宽度。为了便于读者使用，还将修改后的 adodb-pager.ini.php 从 ADODB 类库中复制到实例的根目录下。

实例 543 ADODB 操作结果集的方法

光盘位置：光盘\MR\13\543

高级

趣味指数：★★★★☆

实例说明

通过 ADODB 类库操作数据库，不但要掌握数据库的连接方法，还要掌握操作结果集的方法。在本实例中将介绍 ADODB 类库中操作结果集的方法，并且应用其中的方法循环输出数据表中的字段和值，其运行结果如图 13.6 所示。



图 13.6 ADODB 操作结果集的方法

关键技术

当使用 execute()函数执行 SQL 指令时，会回传一个 ADORecordSet 对象。通过对 ADORecordSet 对象的控制，可以对结果集进行很多操作。操作结果集的方法如下：

（1）fields

该变量保存的是当前指针所指向的记录。

（2）EOF

该变量记录当前指针是否指向最后一条记录。如果是，则设定为 TRUE，否则为 FALSE。

（3）GetArray([$number_of_rows])

返回从当前指针指向的记录开始，到$number_of_rows−1 行的全部记录的数组。参数$number_of_rows 表示指定的记录行，如果没有给出则一直到 EOF 才停止。

（4）MoveNext()

函数作用：将 ADORecordSet（结果集）的指针下移一位。如果成功则返回 TRUE，否则返回 FALSE。

（5）Move($to)

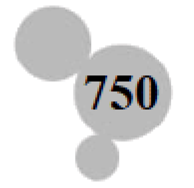
函数作用：将 ADORecordSet（结果集）的指针移动到指定位置。如果$to 等于 0，则指针指向结果集的第一条数据；如果$to 的值大于结果集，则指针指向最后一条数据。注意，这里的变量$to 只能是绝对定位。

（6）MoveFirst()

函数作用：将指针移动到第一条数据，等同于 Move(0)。

（7）MoveLast()

函数作用：将指针移动到最后一条数据，等同于 Move(RecordCount()−1)



```
                    $type = $fields->type;
?>
<td bgcolor="#FFFFFF">
<?php
//从对象中获取字段的类型信息
if ($rst->metaType ( $type, - 1, $fields ) == "T")                    //如果标准类型为 T
        echo $conn->DBDate ( $rst->fields [$i] );                     //使用 DBDate()函数格式化时间
else
        echo $rst->fields [$i];                                       //如果是其他类型，直接输出
?>
</td>
<?php
}
}
?>
```

秘笈心法

心法领悟 543：操作结果集方法的运用。

在本实例中，只是使用 FieldCount()函数和 FetchField()函数对数据表中的字段进行操作。在实例 544 中，将应用 EOF、fields 和 movenext()函数以及 while 语句完成数据表中数据的循环输出，其实现的是结果集中行的操作。

实例 544	ADODB 控制结果集的存取方法 光盘位置：光盘\MR\13\544	高级 趣味指数：★★★★☆

实例说明

在实例 543 中介绍了很多操作结果集的方法，本实例继续应用上一实例中介绍的操作结果集方法，对数据表中的行进行操作，循环输出数据表中的数据。同时应用一个新的方法，实现对结果集存取方式的控制，其运行结果如图 13.7 所示。



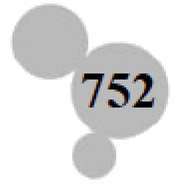
图 13.7　ADODB 类库中对数据库中数据输出的控制

关键技术

在通过对 SQL 语句的操作之后，多数都会返回一个结果集（ADORecordSet 对象）。通过对 ADORecordSet 对象的控制，可以对结果集进行各项操作，这里介绍的是如何控制结果集的存取方法。

1. $ADODB_COUNTRECS

当该变量为 TRUE 时，RecordCount()函数会在数据库驱动不支持 select 指令返回的记录总数时进行自动模



设计过程

（1）创建 conn 文件，编写 conn.php 文件，完成与两个数据库服务器的连接，其代码请参考关键技术中的内容。

（2）创建 index.php 文件，将从 MySQL 数据库中读取出的数据通过 GetInsertSQL()函数添加到 Access 数据库中，并且在本页中输出 MySQL 数据库中的数据，同时创建导出和查看 Aceess 数据库中数据的超链接，其关键代码如下：

```
<?php
include ("conn/conn.php");
$sql = "select * from tb_user where id limit 5 ";                                         //创建 SQL 语句
$rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );               //执行查询语句
$array = array ();
while ( ! $rst->EOF ) {
        $array ['id'] = $rst->fields ['id']; # process $arr
        $array ['user'] = iconv ( "utf-8", "gb2312", $rst->fields ['user'] ); # process $arr
        $array ['pass'] = $rst->fields ['pass']; # process $arr
        $array ['dates'] = $rst->fields ['dates']; # process $arr
        if ($_GET ['insert'] == "导出") {
                $sqls = "select * from tb_user where id=0 ";                              //创建 SQL 语句
                $ret = $con->execute ( $sqls ) or die ( 'error:   ' . $con->errorMsg () ); //执行查询语句
                $insert = $con->getInsertSQL ( $ret, $array ) or die ( 'update error: ' . $con->errorMsg () ); //添加新数据
                $con->execute ( $insert );
        }
?>
<tr>
        <td><?php echo $rst->fields ['id']; ?></td>
        <td height="20"><?php echo $rst->fields ['user']; ?></td>
        <td><?php echo $rst->fields ['dates']; ?></td>
</tr>
<?php
        $rst->movenext ();                                                                //指针下移
}
$rst->close ();                                                                           //关闭连接
$conn->close ();
?>
```

（3）创建 access.php 文件，包含数据库连接文件 conn.php，完成 Access 数据库中数据的循环输出。

秘笈心法

心法领悟 545：通过 ADODB 连接 Access 数据库。

通过 ADODB 连接 Access 数据库时，在设置数据库文件的存储位置时，必须指定数据库文件在服务器中的绝对路径。同时为了 Access 数据库文件的安全，可以为 Access 数据库设置密码。

选择 Access 菜单栏中的“工具”/“安全”/“设置数据库密码”命令可设置 Access 数据库的密码；选择 Access 菜单栏中的“工具”/“安全”/“撤销数据库密码”命令，在弹出的“撤销数据库密码”对话框中输入设置时的密码，即可撤销该数据库的密码。

实例 546	ADODB 连接数据库类 光盘位置：光盘\MR\13\546	高级 趣味指数：★★★☆

实例说明

在 MySQL 数据库中有一个 limit 关键字，通过它可以控制 select 查询语句从第几条记录开始查询及查询多少条记录。而在 ADODB 中也有一个 limit 关键字，就是 SelectLimit()函数，它模拟 limit 关键字，控制 select 查询语句的执行。在本实例中就应用 SelectLimit()函数，查询班级中前 3 名学生的成绩，其运行结果如图 13.10 所示。

秘笈心法

心法领悟 546：明确 SelectLimit()函数与 MySQL 数据库中 limit 关键字的区别。

在 SelectLimit()函数中，$numrows 指定的是要查询的记录数，$offset 指定从第几条记录开始查询，而在 MySQL 数据库中，limit 关键字指定查询的开始位置，而 offset 指定查询的记录数。

实例 547	ADODB 生成 HTML 表格 光盘位置：光盘\MR\13\547	高级 趣味指数：★★★★☆

实例说明

在 ADODB 类库中，有一种方法可以返回表格式的结果集，即将查询结果直接以表格的形式进行输出。本实例就应用这种方法循环输出编程词典的数据，其运行结果如图 13.11 所示。



序号	名称	类别	时间
1	VC编程词典	标准版	Thu 29, Jul 2010
2	VB编程词典	标准版	Thu 29, Jul 2010
3	JAVA编程词典	标准版	Thu 29, Jul 2010

图 13.11　ADODB 生成 HTML 表格

关键技术

本实例应用 rs2html()函数返回一个 HTML 表格格式的结果集，其语法如下：

```
function rs2html($adorecordset,[$tableheader_attributes], [$col_titles])
```

参数说明：

$adorecordset：要返回的结果集。

$tableheader_attributes：对表格的参数及属性的设置。

$col_titles：对字段的重新命名。

要运用本函数，必须载入 ADODB 类库中的 tohtml.inc.php 文件。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）创建 index.php 文件，包含数据库连接文件，定义 SQL 查询语句，应用 rs2html()函数将查询结果以表格的形式输出，其关键代码如下：

```
<?php
include_once 'conn/conn.php';                                   //载入数据库连接文件
include_once '../adodb5/tohtml.inc.php';                        //载入 tohtml.inc.php 文件
$rst = $conn -> execute('select * from tb_bccd');               //返回查询结果集
rs2html($rst,' width="775" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#CCCCCC"',array('序号','名称','类别','时间
'));
?>
```

秘笈心法

心法领悟 547：rs2html()函数。

这是一个独立的函数（rs2heml = recordset to html），相当于 PHP 中的 odbc_result_all()函数。它输出一个完

在本实例中运用的并不是关键技术中讲解的 GetMenu()函数而是 GetMenu2()函数。GetMenu2()函数的作用与 GetMenu()函数相同，它们之间的区别是：GetMenu()中的参数$default_str 和 fields[0]比较，如果相等，那么 fields[0]就是这个下拉列表框的默认选项；而 GetMenu2()中的参数$default_str 则是和 fields[1]进行比较，其他不变。

实例 549　事务处理机制保证数据库操作的完整性

光盘位置：光盘\MR\13\549

高级

趣味指数：★★★★☆

实例说明

事务处理机制在程序开发过程中有非常重要的作用，它可以使整个系统更加安全，例如在银行处理转账业务时，如果 A 账户中的金额刚被发出，而 B 账户还没来得及接受就发生停电事故，这会给银行和个人带来很大的经济损失。采用事务处理机制一旦在转账过程中发生意外，则程序将回滚不做任何处理。在本实例中，沿用实例 545 的内容，实现数据在不同数据库间的转换功能，在将 MySQL 数据库中的数据转换到 Access 数据库中后，删除 MySQL 数据库中的原始数据，并应用事务处理机制保证数据转换操作安全、顺利地进行。运行本实例，如图 13.13 所示，单击“数据转移”超链接，将 MySQL 数据库中的数据转移到 Access 数据库中，并且删除 MySQL 数据库中的原始数据。单击“查看 ACCESS”超链接，将查看到 Access 数据库中存储的数据，如图 13.14 所示。



图 13.13　数据转移



图 13.14　查看 Access 数据库中的数据

关键技术

在 ADODB 中，事务处理应用的是如下 3 个函数：

（1）BeginTrans()函数，开启事务处理。该函数和下面的两个函数都是应用在 PHP 的事务处理中。

（2）CommitTrans()函数，如果成功完成数据库操作则执行该函数。

（3）RollbackTrans()函数，结束一次操作，恢复操作前的所有改变。如果成功返回 TRUE，如果服务器不支持则返回 FALSE。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）编写 index.php 文件，包含数据库连接文件，定义 SQL 语句，循环输出 MySQL 数据库中的数据，通过 BeginTrans()函数开启事务处理。然后根据超链接传递的参数进行判断，通过 GetInsertSQL()函数执行数据的转移操作，转移成功后通过 delete 语句删除原始数据库中的数据。最后根据添加语句和删除语句的执行情况，

判断是执行 CommitTrans ()提交操作还是执行 RollbackTrans ()回滚操作，其关键代码如下：

```
<?php
include ("conn/conn.php");
$sql = "select * from tb_user where id order by id limit 5 ";                     //创建 SQL 语句
$rst = $conn->execute ( $sql ) or die ( 'error：  ' . $conn->errorMsg () );        //执行查询语句
$array = array ();
$con->BeginTrans ();                                                               //开始事务处理
$conn->BeginTrans ();                                                              //开始事务处理
while ( ! $rst->EOF ) {
        if ($_GET ['insert'] == "数据转移") {
                $array ['id'] = $rst->fields ['id']; # process $arr
                $array ['user'] = iconv ( "utf-8", "gb2312", $rst->fields ['user'] ); # process $arr
                $array ['pass'] = $rst->fields ['pass']; # process $arr
                $array ['dates'] = $rst->fields ['dates']; # process $arr
                $sqls = "select * from tb_user where id=0 ";                       //创建 SQL 语句
                $ret = $con->execute ( $sqls ) or die ( 'error：  ' . $con->errorMsg () );   //执行查询语句
                $insert = $con->getInsertSQL ( $ret, $array ) or die ( 'update error：  ' . $con->errorMsg () );   //添加新数据
                $result=$con->execute ( $insert );
                $sql = 'delete from tb_user where id =' . $rst->fields ['id'];     //SQL 删除语句
                $results = $conn->execute ( $sql ) or die ( 'execute error:  ' . $conn->ErrorMsg () );   //执行删除语句
        }
?>
                <tr>
                        <td><?php echo $rst->fields ['id']; ?></td>
                        <td height="20"><?php echo $rst->fields ['user']; ?></td>
                        <td><?php echo $rst->fields ['dates'];     ?></td>
                </tr>
<?php
        $rst->movenext ();                                                         //指针下移
}
if($result){
        $con->CommitTrans ();                                                      //执行提交
        if ($results) {
                $conn->CommitTrans ();                                             //执行提交
        } else {
                $conn->RollbackTrans ();                                           //执行回滚操作
        }
}else{
        $con->RollbackTrans ();                                                    //执行回滚操作
}
$rst->close ();                                                                    //关闭连接
$conn->close ();
?>
```

（3）创建 access.php 文件，循环输出 Access 数据库中存储的数据。

秘笈心法

心法领悟 549：什么是事务。

事务是指一个或一系列的查询，这些查询或者全部执行或者全部不执行，其作用是在服务器发生错误或崩溃的情况下确保数据库的一致性。

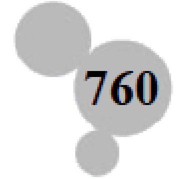
一个事务被永久地写入到数据库中称为事务提交，将状态重置到事务开始之前的状态称为事务回滚。事务具有以下 4 个特性：

（1）原子性：指事务作为一个整体或者完全执行，或者完全不执行。

（2）一致性：指一个事务必须能够使数据处于一致的状态。

（3）孤立性：在事务完全完成之前，它们都是孤立的。

（4）持续性：一旦写入数据库后，事务必须是永久的、持续的。



13.3　ADODB 中的错误处理

实例 550	输出系统中执行的 SQL 语句 光盘位置：光盘\MR\13\550	高级 趣味指数：★★★★☆

实例说明

ADODB 提供很多操作数据库的方法和函数，这其中不乏错误处理机制和程序调试的方法。在本实例中将介绍一种程序调试的方法。通过该方法将输出程序中执行的 SQL 语句，如果 SQL 语句编写错误，那么将输出错误信息，其运行效果如图 13.15 所示。

```
(mysql): select * from tb_use where id limit 5

1146: Table 'db_database13.tb_use' doesn't exist
ADOConnection._Execute(select * from tb_use where id limit 5 , false) % line 1015, file: adodb.inc.php
ADOConnection.Execute(select * from tb_use where id limit 5 ) % line   40, file: index.php
Fatal error: Call to a member function movenext() on a non-object in F:\AppServ\www\MR\13\013\index.php on line 57
```

图 13.15　输出 SQL 语句中的错误信息

关键技术

输出系统中执行的 SQL 语句，应用的是 ADODB 中的 debug 变量。如果变量被设定为 TRUE，当有输出操作时，同时也输出调试信息；如果启用 debug 变量，当错误发生时，将自动调用 ErrorMsg()函数返回错误信息。

ErrorMsg()函数返回最后的状态或出错信息，即使没有错误发生，也会返回一个字符串。所以，一般情况下，只有在发生错误时（返回 FALSE）才调用该函数。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）编写 index.php 文件。首先包含数据库连接文件，设置 debug 变量的值为 TRUE，开启调试功能。然后定义 SQL 语句，循环输出 MySQL 数据库中的数据。最后关闭连接和数据库，其关键代码如下：

```
<?php
	include ("conn/conn.php");
	$conn -> debug = true;						//开启调试
	$sql = "select * from tb_use where id limit 5 ";		//创建 SQL 语句
	$ret = $conn->execute ( $sql ) ;				//执行查询语句
	while ( ! $ret->EOF ) {					//循环输出
?>
<tr>
	<td height="20"><?php echo $ret->fields[0];	?></td>
	<td height="20"><?php echo $ret->fields[1];	?></td>
	<td><?php echo $ret->fields[3];	?></td>
</tr>
<?php
	$ret->movenext ();					//指针下移
}
$ret->close ();							//关闭连接
$conn->close ();
?>
```

秘笈心法

心法领悟 550：合理地运用 debug 调试功能。

debug 变量的调试功能适合在程序开发过程中运用，通过它可以及时发现 SQL 语句中的错误。但是在实际的应用环境中，必须关闭该调试功能。为了确保程序数据库的安全，如果存在 SQL 语句的错误，最好不要让浏览者看到。

实例 551	ErrorMsg 返回错误信息 光盘位置：光盘\MR\13\551	高级 趣味指数：★★★☆

实例说明

本实例将介绍 ADODB 类库中另外一种程序调试的方法，该方法可以在程序发生错误时返回错误信息，其效果如图 13.16 所示。

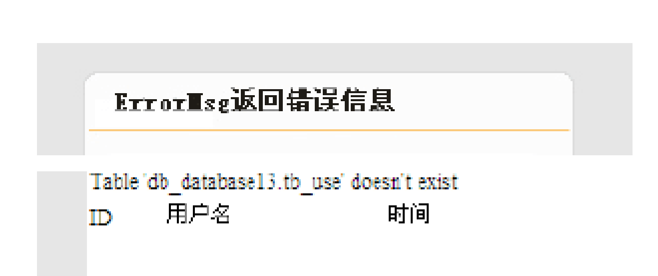


图 13.16　返回错误信息

关键技术

ErrorMsg()函数返回最后的状态或出错信息，即使没有错误发生也会返回一个字符串。所以多数情况下，其与 die()函数联合使用，在执行的 SQL 语句发生错误时（返回 FALSE）才调用该函数。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

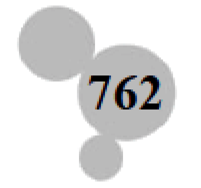
（2）编写 index.php 文件。首先包含数据库连接文件。然后定义 SQL 语句，通过 execute()函数执行 SQL 语句，应用 die()函数在 SQL 语句的执行发生错误时，调用 ErrorMsg()函数返回错误信息。最后循环输出 MySQL 数据库中的数据，其关键代码如下：

```
<?php
	include ("conn/conn.php");
	$sql = "select * from tb_use where id limit 5 ";				//创建 SQL 语句
	$ret = $conn -> execute($sql) or die($conn -> ErrorMsg());		//执行查询语句并返回错误信息
	while ( ! $ret->EOF ) {								//循环输出
?>
<tr>
	<td height="20"><?php echo $ret->fields[0];	?></td>
	<td height="20"><?php echo $ret->fields[1];	?></td>
	<td><?php echo $ret->fields[3];	?></td>
</tr>
<?php
	$ret->movenext ();								//指针下移
}
$ret->close ();										//关闭连接
$conn->close ();
?>
```

秘笈心法

心法领悟 551：合理地运用 ErrorMsg()函数返回 SQL 语句的错误信息。

ErrorMsg()函数返回最后的状态或出错信息，其特点是无论 SQL 语句是否出错都会返回一个字符串，这就要求在应用 ErrorMsg()函数时必须注意。为避免在 SQL 语句执行正确的情况下返回字符串，在 SQL 语句的执行发生错误时运用 ErrorMsg()函数是最佳方案。



秘笈心法

心法领悟 553：ADODB_ERROR_LOG_TYPE 常量设置错误记录的类型。

用 ADODB_ERROR_LOG_TYPE 常量设置错误记录的类型，可选值如下：

☑　0：默认值。根据 php.ini 文件中的 error_log 配置，错误被发送到服务器日志系统或文件。

☑　1：错误被发送到 destination 参数中的地址。只有该类型使用 headers 参数。

☑　2：通过 PHP debugging 连接来发送错误。该选项只在 PHP 3 中可用。

☑　3：错误被发送到文件目标字符串。

13.4　ADODB 操作数据库

实例 554	应用 ADODB 访问 Access 数据库 光盘位置：光盘\MR\13\554	高级 趣味指数：★★★☆

实例说明

虽然 PHP 是构建 Web 系统强有力的工具，但是由于 PHP 存取函数一直没有标准化，不同数据库间的函数名称、参数差异很大，在更换数据库时，会带来大量的代码修复工作。这时，就需要一组函数库来隐藏不同数据库函数间的差异，使开发者可以很简单地去切换数据库，这就是 ADODB 类库。

本实例中，应用 ADODB 访问 Access 数据库，并通过分页函数实现数据库中数据的分页输出，其运行结果如图 13.19 所示。



图 13.19　ADODB 分页输出 Access 数据库中的数据

关键技术

应用ADODB操作Access数据库最简单的方法就是应用实例538和实例539中定义的ADODB连接和操作数据库类。本实例的目的是使读者更深刻地理解 ADODB 操作 Access 数据库的方法和步骤。

（1）载入 adodb.inc.php 文件，这是应用 ADODB 类库操作任何数据库的前提。

（2）连接 Access 数据库服务器，依然应用 ADONewConnection()函数，但是其参数值是 Access。

（3）应用 PConnect()函数连接指定的数据库文件。这是 Access 数据库与其他数据库不同的地方，数据库文件的位置必须是一个绝对路径。

（4）设置数据库的编码格式。

ADODB 连接 Access 数据库的关键代码如下：

```
<?php
include('../adodb5/adodb.inc.php');                          //载入（include）adodb.inc.php 文件
$conn = ADONewConnection('access');                          //连接 Access 数据库
$conn-> PConnect("Driver={Microsoft Access Driver (*.mdb)};Dbq=F:\\AppServ\\www\\mr\\13\\017\\data\\db  database13.mdb");
$conn -> execute('set names gb2312');                        //设置编码格式
?>
```

图 13.20 所示。



图 13.20　ADODB 访问 MySQL 数据库

关键技术

在本实例中，应用 ADONewConnection()函数连接数据库服务器，应用 PConnect()函数连接 db_database13 数据库，设置用户名为 root，密码为 111，设置数据库编码格式为 UTF-8。

通过 execute()函数执行 SQL 语句，在 while 循环语句中，应用 EOF 判断指针位置，应用 fields 获取数据表中对应字段的值，应用 MoveNext()函数控制指针的移动。

本实例中应用的函数请读者参考实例 538 和实例 543 关键技术中的内容，这里不再赘述。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）创建 index.php 文件。首先包含数据库连接文件，然后定义 SQL 语句，通过 execute()函数执行 SQL 语句，最后通过 while 循环输出 MySQL 数据库中的数据，其关键代码如下：

```
<?php
include_once ('conn/conn.php');                          //载入（include）adodb.inc.php 文件
$sqlstr = 'select * from tb_bccd';                        //创建 SQL 语句
$rst = $conn -> execute($sqlstr) or die('connect error');  //执行 SQL 语句，查询出 id 等于 2 的数据
while(!$rst -> EOF){                                      // while 语句循环输出结果
?>
  <tr>
    <td   height="20" bgcolor="#FFFFFF"><?php echo $rst -> fields['id'];?></td>
    <td bgcolor="#FFFFFF"><?php echo $rst -> fields['name'];?></td>
    <td bgcolor="#FFFFFF"><?php echo $rst -> fields['type'];?></td>
    <td bgcolor="#FFFFFF"><?php echo $rst -> fields['dates'];?></td>
  </tr>
<?php
        $rst -> MoveNext();                               //指针下移
}
$rst -> close();                                          //关闭连接
$conn -> close();                                         //关闭数据库
?>
```

秘笈心法

心法领悟 555：选择适合的 ADODB 操作数据库方法。

在实际的 Web 程序开发过程中，建议读者应用实例 538 和实例 539 中封装的 ADODB 连接和操作数据库类来实现对数据库的操作。因为这种方法更适合对数据库的管理、维护和更新，一旦出现更换数据库服务器的问题也容易解决。本实例中介绍的单一的数据库连接、操作方法存在一定的局限性，更适合于读者在学习阶段使

用，以了解 ADODB 操作数据库的函数和步骤。

实例 556	应用 ADODB 访问 SQL Server 数据库 光盘位置：光盘\MR\13\556	高级 趣味指数：★★★★☆

实例说明

本实例介绍应用 ADODB 访问 SQL Server 数据库，并且通过 while 循环语句输出数据库中的数据，其中还应用$ADODB_FETCH_MODE 变量控制结果集的存储方式，运行结果如图 13.21 所示。



图 13.21　ADODB 访问 SQL Server 数据库

关键技术

通过 ADODB 访问 SQL Server 数据库的操作步骤与实例 555 应用 ADODB 访问 MySQL 数据库是相同的，这里不再赘述，唯一的区别是在访问 SQL Server 数据库时，应用$ADODB_FETCH_MODE 变量对结果集的存取方式进行控制。

$ADODB_FETCH_MODE 变量决定结果集以哪种方式进行存取，有关该变量的详细讲解请参考实例 544。

在通过 ADODB 连接 SQL Server 数据库时还有一点需要注意，即在通过 PConnect()函数连接数据库文件时，对 Server 服务器的设置。如果 SQL Server 数据库存储于本机中，那么 Server 的参数值可以使用 localhost。但是，如果 SQL Server 数据库存储在其他机器中，那么 Server 的参数值必须是 SQL Server 的服务器名称。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 SQL Server 数据库的连接，并且设置以字段名称为结果集索引。

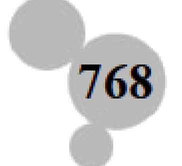
（2）编写 index.php 文件。首先包含数据库连接文件，然后定义 SQL 语句，通过 execute()函数执行 SQL 语句，最后循环输出 SQL Server 数据库中的数据，其关键代码如下：

```
<?php
include_once("conn/conn.php");
$sqlstr = "select * from tb_goods where addtime='2009-5-23' ";          //创建 SQL 语句
$rst = $conn -> execute($sqlstr) or die('connect error');              //执行 SQL
while(!$rst -> EOF){                                                  //如果没有错误，则配合 while 循环输出结果
?>
    <tr>
      <td align="left" bgcolor="#FFFFFF"> <?php echo iconv('gb2312','utf-8',$rst -> fields['name']);?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $rst -> fields['addtime'];?></td>
      <td align="left" bgcolor="#FFFFFF"> <?php echo $rst -> fields['counts'];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $rst -> fields['price'];?></td>
    </tr>
<?php
      $rst -> movenext();                                             //指针下移
}
$rst -> close();                                                      //关闭连接
$conn -> close();
?>
```

秘笈心法

心法领悟 556：了解 ADODB 操作数据库的方法和步骤。

在本节的 3 个实例中，分别介绍了如何通过 ADODB 操作 Access、MySQL 和 SQL Server 数据库，其目的是让读者对 ADODB 中操作数据库的方法和步骤有一个系统的了解，从中读者可以体会到，无论是操作哪种数据库，所使用的方法和操作步骤基本都是相同的，这就是 ADODB 类库的强大之处，这也就是笔者建议在实际的程序开发过程中使用实例 538 和实例 539 中封装的类来连接和操作数据库的原因。



14.1　环 境 搭 建

实例 557	Smarty 开发环境搭建 光盘位置：光盘\MR\14\557	高级 趣味指数：★★★★☆

实例说明

PHP 没有内置 Smarty 模板类，需要单独下载和配置，并且 Smarty 要求服务器上的 PHP 版本最低为 4.0.6。用户可以去网站 http://www.smarty.net/download.php 下载最新的 Smarty 压缩包。本书使用的版本是 Smarty 2.6.23，下载页面如图 14.1 所示。

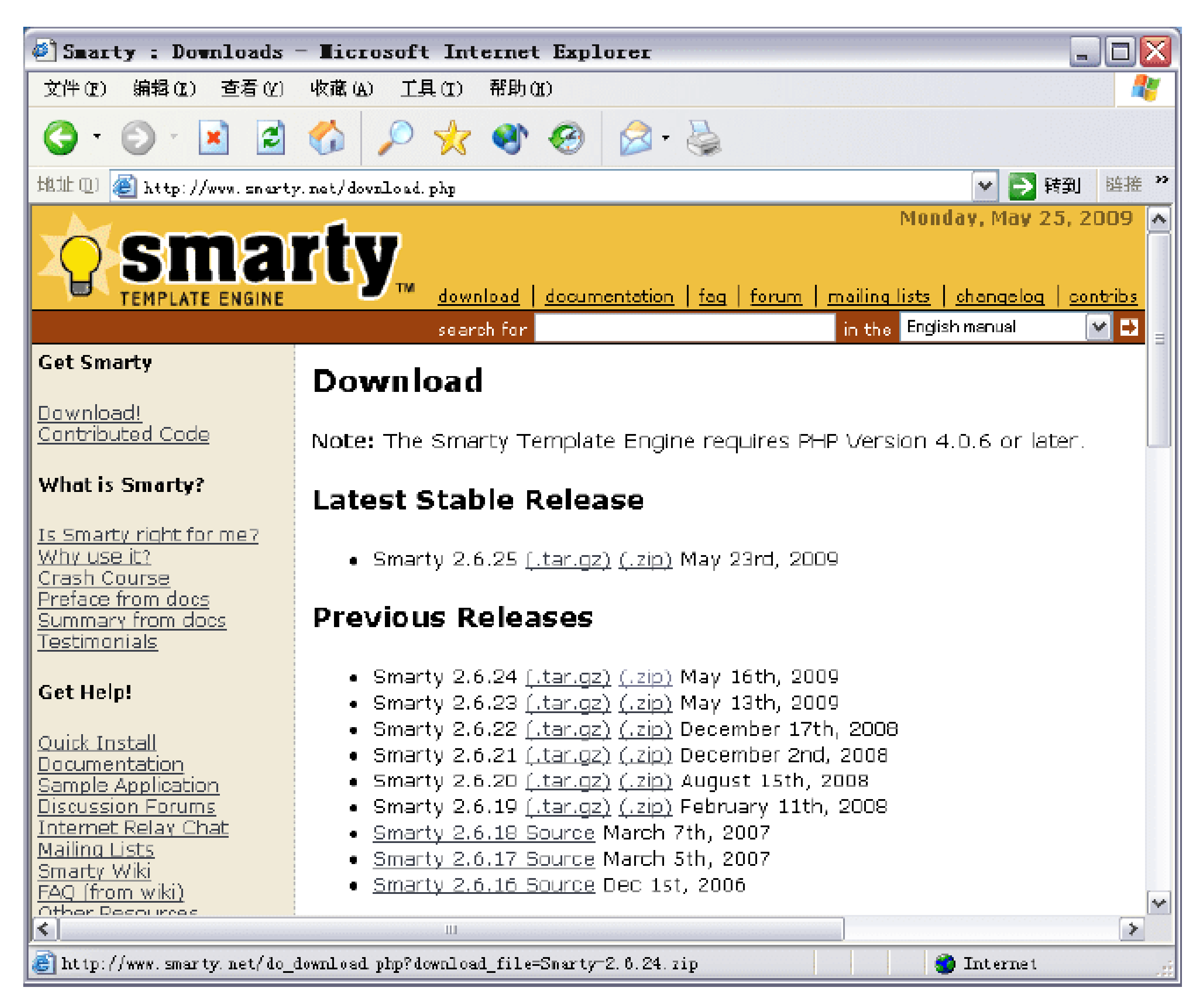


图 14.1　Smarty 模板下载页面

关键技术

Smarty 模块的安装步骤：

（1）将下载的 Smarty 压缩包解压，解压后的文件夹如图 14.2 所示，其中有一个 libs 文件夹，它包含了 Smarty 类库的 4 个核心文件：Smarty.class.php、Smarty_Compiler.class.php、Config_File.class.php 和 debug.tpl，以及 internals 和 plugins 两个文件夹，如图 14.3 所示。

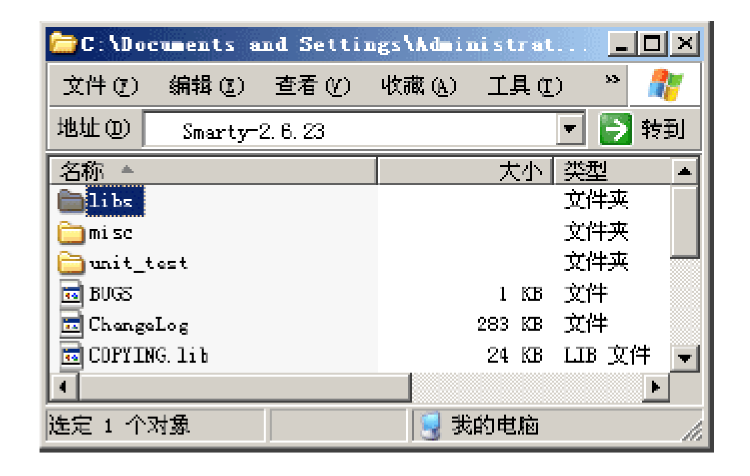
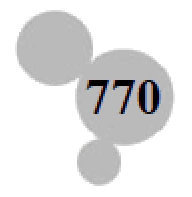


图 14.2　Smarty-2.6.23 的压缩包



图 14.3　Smarty-2.6.23 的 libs 文件



件。这 4 个文件夹具体放置在什么位置没有严格的规定，只要设置路径正确即可。在本实例中，将创建的 4 个文件夹存储在 558\Smarty 文件夹下。



图 14.6 第一个 Smarty 程序

（3）创建配置文件。如果要应用 Smarty 模板，那么就一定要包含 Smarty 类库和相关信息。配置信息可以存储在一个单独的文件中，在需要时通过 include 语句包含即可。配置文件 config.php 的代码如下：

```
<?php
	/*  定义服务器的绝对路径  */
	define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);
	/*  定义 Smarty 模板的绝对路径  */
	define('SMARTY_PATH','\mr\14\Smarty\\');
	/*  加载 Smarty 类库文件  */
	require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
	/*  实例化一个 Smarty 对象  */
	$smarty = new Smarty;
	/*  定义各个目录的路径  */
	$smarty->template_dir = './';                               //定义模板文件的存储位置
	$smarty->compile_dir = './Smarty/templates_c/';             //定义编译文件的存储位置
	$smarty->config_dir = './Smarty/configs/';                  //定义配置文件的存储位置
	$smarty->cache_dir = './Smarty/cache/';                     //定义缓存文件的存储位置
?>
```

设计过程

完成 Smarty 模板的配置后，编写一个实例，体会 Smarty 模板引擎的特性。

（1）在实例 558 文件夹下创建 index.php 文件，编写动态 PHP 代码。首先设置页面的编码格式，然后通过 include 语句包含 Smarty 的配置文件，最后通过 Smarty 类库中的 assign()方法将数据赋给模板变量，并且应用 display 指定模板页，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );            //设置页面编码格式
include("config.php");                                          //包含配置文件
/*  使用 Smarty 赋值方法将一对名称/方法发送到模板中  */
$smarty->assign('title','第一个 Smarty 程序');
$str = file_get_contents('files/content.txt');                  //读取文本文件中的数据
$smarty->assign('content',iconv("gb2312","utf-8",$str));        //将文本文件中的数据存储到模板变量中
/*  显示模板  */
$smarty->display('index.html');
?>
```

☑ assign()方法用于在模板被执行时为模板变量赋值，语法如下：

```
{assign var=" " value=" "}
```





图 14.7　找不到指定的 Smarty 类库文件

本实例向读者介绍另外一种 Smarty 模板的配置方法，使用这种方法，无论怎么改变实例的文件夹，都不会影响实例的运行。

关键技术

这里将 Smarty 配置方法封装到类中，并存储到 system.smarty.inc.php 文件中。首先包含 Smarty 类文件 Smarty.class.php，然后定义 SmartyProject 类，继承 Smarty 父类，最后定义 SmartyProject()方法，设置 Smarty 中模板文件（templates）、编译文件（templates_c）、配置文件（configs）和缓存文件（cache）的存储位置。这就是新的配置文件 system.smarty.inc.php，其代码如下：

```
<?php
require("../Smarty/Smarty.class.php");                    //调用 Smarty 文件
class SmartyProject extends   Smarty{                     //定义类，继承 Smarty 父类
 function SmartyProject(){                                //定义方法，配置 Smarty 模板
        $this->template_dir = "./";                       //指定模板文件存储在根目录下
        $this->compile_dir = "./system/Smarty/templates_c/";   //指定编译文件存储位置
        $this->config_dir = "./system/Smarty/configs/";
        $this->cache_dir = "./system/Smarty/cache/";
    }
}
?>
```

既然已经将 Smarty 的配置方法存储到一个类中，那么就需要对类进行实例化，根据返回的对象名称调用 Smarty 中的方法，类的实例化操作在 system.inc.php 文件中完成，其返回对象名为$smarty，其代码如下：

```
<?php
 require("system.smarty.inc.php");                        //调用类文件
 $smarty=new SmartyProject();                             //执行类的实例化操作
?>
```

通过此方法配置 Smarty 模板的好处是，无论将程序复制到哪个服务器下执行，都不需要更改服务器或者 Smarty 文件的绝对路径，程序可以直接运行。

✍ **技巧：** 这种配置方法中调用的仍是存储在章文件夹 14 下的 Smarty 类库，同时在这种方法中将配置文件存储于实例根目录下的 system 文件夹中，将模板文件夹（templates）、编译文件夹（templates_c）、配置文件夹（configs）和缓存文件夹（cache）存储在 system\Smarty 目录下。

设计过程

下面应用新的配置方法对实例 558 进行重新编写，生成实例 559，步骤如下：

（1）在实例的根目录下创建 system 文件夹，存储 Smarty 的配置文件 system.smarty.inc.php 和实例化操作文件 system.inc.php，同时创建 Smarty 文件夹，最后在 Smarty 文件夹下创建 3 个文件夹 templates_c、configs 和 cache。

（2）编写 index.php 文件，重新载入 Smarty 的配置文件，即通过 include 语句包含 system.inc.php。

（3）其他内容与实例 558 相同，这里不再赘述。



关键技术

在模板页中判断用户访问权限的关键是 if 语句和模板变量的结合运用。在 if 语句中，判断模板变量的值是否为 T，如果为 T，则说明具备访问权限；如果模板变量值为 F，则说明不具备访问权限。

Smarty 模板中的 if 与 PHP 中的 if 大同小异。需要注意的一点是，Smarty 模板中的 if 必须以/if 为结束标记，其语法格式如下：

```
{if 条件语句 1}
语句 1
{elseif 条件语句 2}
语句 2
{else}
语句 3
{/if}
```

模板变量的定义应用 assign()方法，在 PHP 动态页中根据用户权限的判断结果为模板变量赋值。

设计过程

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，创建存储编译文件、缓存文件和配置文件的文件夹。其具体内容可以参考实例 559，这里不再赘述。

（2）创建 index.php 文件，包含 Smarty 配置文件，指定 Smarty 的模板页 index.html。

（3）创建 Smarty 模板页 index.html，设计用户登录页面，添加用户登录的表单元素，并且将用户的登录信息提交到 index_ok.php 文件。

（4）创建 index_ok.php 文件，获取表单中提交的数据，判断提交的用户名和密码是否正确，如果正确则通过 display()方法指定模板页 main.html，并且为模板变量 competence 赋值为 T；否则指定到模板页 main.html，为模板变量 competence 赋值为 F，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include("system/system.inc.php");                                  //包含配置文件
if($_POST['Submit']!="" && $_POST['user']!="" && $_POST['pass']!=""){
        if($_POST['user']=="mr" && $_POST['pass']=="mrsoft"){
                $smarty->assign("competence","T");                  //为模板变量赋值
                $smarty->display('main.html');                      //指定模板页
        }else{
                $smarty->assign("competence","F");
                $smarty->display('main.html');
        }
}else{
        echo "<script>alert('用户名和密码不能为空！'); window.location.href='index.php';</script>";
}
?>
```

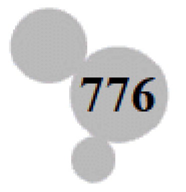
（5）创建 main.html 模板页。首先应用 if 语句对模板变量的值进行判断，如果值为 T 则输出本页内容；如果值为 F 则给出提示信息，其关键代码如下：

```
{if $competence=="T"}
//省略了部分内容
{/if}
{if $competence=="F"}
<script>alert('您没有权限访问，请重新登录');window.location.href="index.php";</script>
{/if}
```

秘笈心法

心法领悟 560：Smarty 模板中 if 条件语句的修饰词。

Smarty 模板中 if 条件语句，除了可以使用 PHP 中的<、>、=、!=等常见运算符外，还可以使用 eq、ne、neq、gt、lt、lte、le、gte、ge、is even、is odd、is not even、is not odd、not、mod、div by、even by、odd by 等修饰词修饰。



实例 561　Smarty 模板中生成数字验证码

光盘位置：光盘\MR\14\561

高级

趣味指数：★★★☆

实例说明

本实例应用 Smarty 模板中的 foreach 语句在模板页中直接生成一个数字验证码，它相当于在动态 PHP 中开发的验证码，但更加简单、实用。Smarty 模板中生成验证码的运行结果如图 14.10 所示。



图 14.10　Smarty 模板中生成数字验证码

关键技术

严格地说，这个验证码不是在 Smarty 模板页中生成的，只是通过 foreach 语句将 PHP 动态页中生成的验证码在模板页中输出。因此本实例开发的关键是 foreach 语句的应用和验证码的生成。

（1）应用 mt_rand()函数生成验证码，其语法如下：

```
int mt_rand ( [int min, int max])
```

如果 mt_rand()函数没有提供可选参数 min 和 max，则返回 0~RAND_MAX 之间的伪随机数。

（2）应用 explode()函数将生成的验证码写入数组中，其语法如下：

```
array explode(string separator, string string, [int limit])
```

返回由字符串组成的数组，每个元素都是 string 的一个子串，它们被字符串 separator 作为边界点分隔出来。如果设置 limit 参数，则返回的数组包含最多 limit 个元素，而最后那个元素将包含 string 的剩余部分。

如果 separator 为空字符串（""），explode()函数将返回 FALSE；如果 separator 所包含的值在 string 中找不到，那么 explode()函数将返回包含 string 单个元素的数组；如果参数 limit 是负数，则返回除了最后的-limit 个元素外的所有元素。

（3）通过 assign()方法将验证码数组赋给模板变量。

（4）应用 foreach 语句循环输出模板变量中的数组元素。foreach 的语法如下：

```
{foreach name=foreach_name key=key item=item from=arr_name}
…
{/foreach}
```

参数 name 为该循环的名称；key 为当前元素的键值；item 是当前元素的变量名；from 是该循环的数组。其中，item 和 from 是必选参数，不可省略。

设计过程

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，这里不再赘述。

（2）创建 index.php 文件，生成验证码并且将验证码赋给模板变量，最后指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                        //包含配置文件
$array= explode(' ', mt_rand(1000,9999));                     //生成随机验证码
$smarty->assign('title','Smarty 模板中生成数字验证码');           //将指定数据赋给模板变量
$smarty->assign('content',$array);                            //将数组赋给模板变量
$smarty->display('index.html');                               //指定模板页
?>
```

（3）创建 index.html 模板文件，设计用户登录页面，添加用户登录的表单元素，同时应用 foreach 语句输出模板变量中传递的验证码，其关键代码如下：

```
<form id="form1" name="form1" method="post" action="index_ok.php">
        <input name="user" type="text" id="user" size="10" />
        <input type="hidden" name="checks" value="{foreach key=key item=item from=$content} {$item} {/foreach}" />
        <input name="check" type="text" size="8" />
        <input name="pass" type="password" id="pass" size="10" />
        <input type="image" name="imageField" src="images/Blog_03_03.jpg" />
        <input type="image" name="imageField2" onclick="form.reset();return false;" src="images/Blog_03_06.jpg" />
</form>
```

（4）创建 index_ok.php 文件，获取表单中提交的数据，对用户登录信息进行验证。如果用户名和密码正确，则跳转到 main.php 页面，否则跳转到 index.php 页面。

（5）创建 main.php 和 main.html 文件作为网站的主页。

秘笈心法

心法领悟 561：验证码的运用。

本实例中，应用 foreach 语句输出验证码，如果要更新验证码的值，可以单击“刷新”按钮或者按 F5 键，同样也可以在地址栏中重新输入地址。

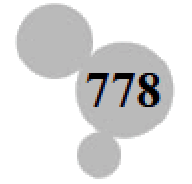
实例 562	Smarty 模板中的页面设计 光盘位置：光盘\MR\14\562	高级 趣味指数：★★★★☆

实例说明

在 Smarty 模板页中，同样可以将网页的头、尾和主文件分别进行存储，从而达到页面重用的效果，这就是本实例将要讲解的内容——在 Smarty 模板页中设计网页页面，其运行结果如图 14.11 所示。



图 14.11　Smarty 模板中的页面设计



关键技术

在 Smarty 模板页中应用 include 函数载入模板页。include 函数用于在当前模板中包含其他模板，当前模板中的变量在被包含的模板中可用。include 函数的语法如下：

```
{include file="file_name " assign=" " var="   "}
```

参数 file 指定包含模板文件的名称，为必选参数；参数 assign 指定一个变量保存包含模板的输出；参数 var 传递待包含模板的本地参数，只在待包含模板中有效。

设计过程

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，这里不再赘述。

（2）创建 index.php 文件，应用 switch 语句根据超链接传递的变量值，包含不同的 PHP 脚本文件，同时将对应的模板页名称赋给指定的模板变量，其代码如下：

```
<?php
require("system/system.inc.php");                    //载入配置文件
switch ($_GET['caption']){                           //应用 switch 语句进行判断
case "reg":
        include "reg.php";                           //包含 PHP 脚本
        $smarty->assign('admin_phtml','reg.html');   //通过模板变量指定模板页
        break;
case "log":
        include "log.php";
        $smarty->assign('admin_phtml','log.html');
        break;
case "from":
        include "for.php";
        $smarty->assign('admin_phtml','for.html');
        break;
case "mes":
        include "mes.php";
        $smarty->assign('admin_phtml','mes.html');
        break;
default:
        include "for.php";
        $smarty->assign('admin_phtml','for.html');
        break;
}
$smarty->assign("title","Smarty 模板中的页面设计--".$_GET[caption]);
$smarty->display("index.html");                      //指定模板页
?>
```

（3）创建 index.html 模板页，应用 include 函数载入页面的头文件（top.html）、尾文件（bottom.html）和主文件，其关键代码如下：

```
{$title}
{include file=top.html}
{include file=$admin_phtml}
{include file=right.html}
{include file=bottom.html}
```

（4）创建 include 函数加载的模板文件和动态 PHP 文件。

秘笈心法

心法领悟 562：switch 语句的应用。

在程序设计中，所有依据条件做出判定的问题，都可以用 if 条件语句来解决。不过，在用 if...else...语句处理多个条件的判定问题时，组成条件的表达式在每一个 elseif 中都要计算一次，比较繁琐。为了避免 if 语句的冗长，提高程序的可读性，可以使用 switch 分支控制语句。switch 语句的语法如下：

```
switch(variable){
case value1:
        statement1;
        break;
```

```
case value2:
…
default:
        default statement n;
}
```

switch 语句根据 variable 的值，依次与 case 中的 value 值相比较，如果不相等，继续查找下一个 case；如果相等，就执行对应的语句，直到 switch 语句结束或遇到 break 为止。一般 switch 语句的最后都有一个默认值 default，如果在前面的 case 中没有找到相符的条件，则输出默认语句，和 else 语句类似。

实例 563　Smarty 模板中直接定义 CSS 样式

光盘位置：光盘\MR\14\563

高级

趣味指数：★★★☆

实例说明

由于在 Smarty 模板中默认使用“{”和“}”作为定界符，所以如果直接在 Smarty 模板中定义 CSS 样式，将输出如图 14.12 所示的错误。

该错误出现的原因是在 Smarty 模板页中直接将 CSS 样式中使用的“{”和“}”解析为 Smarty 模板的定界符，为了避免在 Smarty 模板中直接定义 CSS 样式时出现错误，Smarty 提供了特殊的处理方法，这就是本实例将要讲解的 literal 标签。

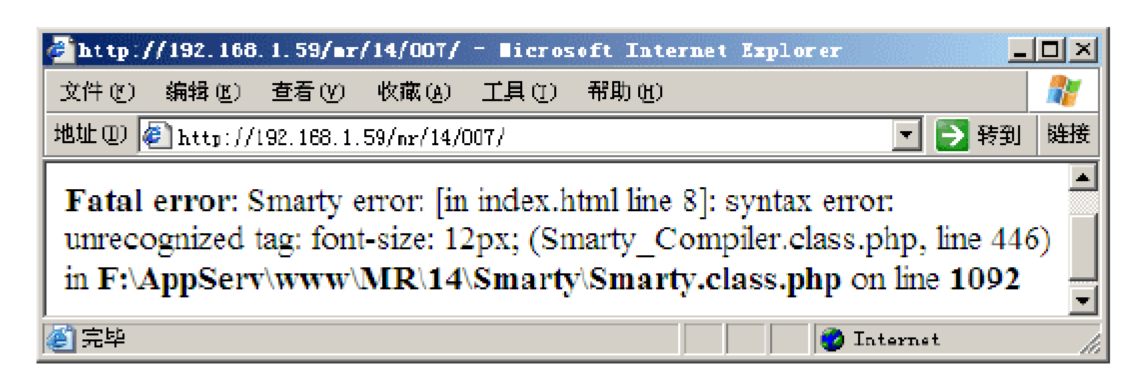


图 14.12　在 Smarty 模板中直接定义 CSS 样式时出现的错误

关键技术

literal 标签将区域内的数据当作文本处理，此时模板将忽略其内部的所有字符信息。该特性用于显示有可能包含大括号等字符信息的 CSS 样式或者 JavaScript 脚本，当这些信息处于 {literal}{/literal} 标签中时，模板引擎将不进行分析而直接显示。

设计过程

这里仍然应用实例 561 中的内容，唯一的区别是在 index.html 模板页中，通过{literal}{/literal} 标签定义一个 CSS 样式，控制页面中表格、文字的背景、字体和大小等，其关键代码如下：

```
{literal}
<style type="text/css">
<!--
body,td,th {
 font-size: 12px;
}
body {
 margin-left: 0px;
 margin-top: 0px;
 margin-right: 0px;
 margin-bottom: 0px;
}
a:link {
 text-decoration: none;
}
a:visited {
 text-decoration: none;
}
a:hover {
 text-decoration: none;
}
a:active {
```

设计过程

这里仍然应用实例 561 中的内容，唯一的区别是在 index.html 模板页中，直接定义 JavaScript 脚本对表单中提交的数据进行判断，通过 ldelim 和 rdelim 标签在 Smarty 模板中输出“{”和“}”，其关键代码如下：

```
<script language=javascript>
function check_form() {ldelim}
 if (form.user.value == '') {ldelim}
                alert('请输入用户名');
        form.user.focus();
        return false;
 {rdelim}
 if (form.pass.value == '') {ldelim}
                alert('请输入密码');
        form.pass.focus();
        return false;
 {rdelim}
 if (form.check.value == '') {ldelim}
                alert('请输入验证码');
        form.check.focus();
        return false;
 {rdelim}
 {rdelim}
</script>
```

本实例中的其他内容这里不再赘述，其完整代码请参考本书光盘。

秘笈心法

心法领悟 564：literal 标签与 ldelim、rdelim 标签的比较。

literal 标签将“{”与“}”之间的内容按普通的文本文件来解析，控制的是内容；而 ldelim 与 rdelim 标签在 Smarty 模板中输出“{”与“}”，是对大括号本身进行操作。在 Smarty 模板中完成 CSS 样式或者 JavaScript 脚本的嵌入操作时，笔者个人认为还是使用 literal 标签方便一些。

实例 565 html_options 函数向下拉列表框中添加列表项

光盘位置：光盘\MR\14\565

高级

趣味指数：★★★☆

实例说明

在开发程序的过程中，经常用到下拉列表框，可以直接在页面中为其定义值，也可以使用从数据库中读取的数据来定义，但无论使用哪种方法都必须应用 option 标签，这是在常规的 Web 页面中为下拉列表框赋值的方法。

本实例将介绍通过 Smarty 模板中的自定义函数 html_options 为下拉列表框赋值。使用 Smarty 模板中的自定义函数为下拉列表框赋值与使用 option 标签实现的效果是相同的，但是这种方法更适合从数据库中读取数据，更符合 Smarty 开发模式动静分离的原则，运行结果如图 14.14 所示。



图 14.14　通过 html_options 函数为下拉列表框赋值

关键技术

自定义函数 html_options 根据给定的数据创建选项组。该函数可以指定哪些元素被选定，要么指定 values 和 ouput 属性，要么指定 options 替代。自定义函数的语法如下：



```
<select name=customer_id>
 {html_options values=$cust_ids selected=$customer_id output=$cust_names}
</select>
```

html_options 函数的参数说明如表 14.1 所示。

表 14.1　html_options 函数的参数说明

参　　数	类　　型	说　　明
name	string	下拉菜单名称，默认值为空
options	associative array	包含值和显示的关联数组
selected	string/array	已选定的元素或元素数组
output	array	包含下拉列表各元素显示值的数组
values	array	包含下拉列表各元素值的数组

如果给定值是数组，将作为 OPTGROUP 处理，且支持递归。所有的输出与 XHTML 兼容。

如果指定可选属性 name，该选项列表将被置于<select name="groupname"></select>标签中，如果没有指定，只产生选项列表。

设计过程

（1）创建 system 文件夹。首先定义 system.smarty.inc.php 文件，封装 Smarty 的配置方法以及 ADODB 连接和操作数据库的方法。然后定义 system.inc.php 文件，对 Smarty 配置类、数据库的连接和操作类进行实例化，并返回连接对象。最后创建 Smarty 文件夹，定义 Smarty 的编译文件、配置文件和缓存文件的存储目录。

（2）创建 index.php 文件。首先设置页面的编码格式、载入配置文件。然后生成随机验证码，并且将随机验证码的值赋给模板变量。接着从数据库中读取数据，将读取的数据存储到数组中，并且将数组赋给模板变量，作为下拉列表框中的值。最后指定模板页，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once("system/system.inc.php");                                  //包含配置文件
$array= explode(' ', mt_rand(1000,9999));                               //生成随机验证码
$smarty->assign('title','html_option 函数向下拉列表中添加列表项');      //将指定数据赋给模板变量
$smarty->assign('content',$array);                                      //将数组赋给模板变量
$res=$conn->execute("select * from tb_bccd ");                          //执行 select 查询语句
$array=$res->GetArray();                                                //返回查询结果
$array_id=array();
$array_name=array();
for($i=0;$i<$res->RecordCount();$i++){
        $array_id[]=$array[$i][id];
        $array_name[]=$array[$i][name];
}
$smarty->assign('cust_id', $array_id);                                  //设置下拉列表框的值
$smarty->assign('cust_name', $array_name);                              //设置列表框的显示数据
$smarty->display('index.html');                                         //指定模板页
?>
```

（3）创建 index.html 模板页。首先创建 form 表单，添加表单元素，将用户的注册信息提交到 index_ok.php 文件中，其中，通过 html_options 自定义函数输出下拉列表框的值，通过 foreach 语句输出验证码的值。然后应用 literal 标签在页面中直接嵌入 CSS 样式和 JavaScript 脚本，对用户提交的信息进行验证，其关键代码如下：

```
<form id="form" name="form" method="post" action="index_ok.php" onsubmit="return check_form(); ">
词典选择：<select name="customer" id="customer">{html_options output=$cust_name values=$cust_id }</select>
验证码：  <input type="hidden" id="checks" name="checks" value="{foreach key=key item=item from=$content} {$item} {/foreach}" />
<input name="check" type="text" id="check" size="8" />{foreach key=key item=item from=$content} {$item} {/foreach}
<input type="image" name="imageField3" src="images/reg_07.jpg" />
<input type="image" name="imageField4" onclick="form.reset();return false;" src="images/reg_09.jpg" />
</form>
```

（4）创建 index_ok.php 文件，获取表单中提交的数据，将数据添加到数据表中，其代码如下：

```
<?php
```

关键技术

在 Smarty 模板中，通过自定义函数 html_select_date 和 html_select_time 生成日期、时间选择器。

html_select_date 函数用于创建日期下拉菜单，可以显示任意年、月、日。html_select_date 函数的语法说明如表 14.2 所示。

表 14.2 html_select_date 函数的语法说明

属 性	类 型	是否必需的	默 认 值	说 明
year_size	string	No	null	如果设置，为标签添加大小属性
all_extra	string	No	null	如果设置，为所有标签添加附加属性
day_extra	string	No	null	如果设置，为标签添加附加属性
month_extra	string	No	null	如果设置，为标签添加附加属性
year_extra	string	No	null	如果设置，为标签添加附加属性
field_order	string	No	MDY	显示区域的顺序
field_separator	string	No	\n	各区域间输出的分隔字符串
month_value_format	string	No	%m	月份值的 strftime 表示方法，默认为 %m
year_size	string	No	null	如果设置，为标签添加大小属性

html_select_time 函数用于创建时间下拉菜单，可以显示任意时、分、秒。html_select_time 函数的语法说明如表 14.3 所示。

表 14.3 html_select_time 函数的语法说明

属 性	类 型	是否必需的	默 认 值	说 明
prefix	string	No	Time_	变量名称前缀
time	timestamp	No	UNIX	使用时间类型（data/time）
display_hours	boolean	No	true	是否显示小时
display_minutes	boolean	No	true	是否显示分钟
display_seconds	boolean	No	true	是否显示秒
display_meridian	boolean	No	true	是否显示正午界（上午/下午）
use_24_hours	boolean	No	true	是否使用 24 小时制
minute_interval	integer	No	1	分钟下拉列表框的间隔
second_interval	integer	No	1	秒钟下拉列表框的间隔
field_array	string	No	n/a	输出值到该值指定的数组
all_extra	string	No	null	如果设置，为标签添加附加属性
hour_extra	string	No	null	如果设置，为标签添加附加属性
minute_extra	string	No	null	如果设置，为标签添加附加属性
second_extra	string	No	null	如果设置，为标签添加附加属性
meridian_extra	string	No	null	如果设置，为标签添加附加属性

设计过程

（1）创建 system 文件夹。首先定义 system.smarty.inc.php 文件，封装 Smarty 的配置方法以及 ADODB 连

14.3　综 合 应 用

实例 567	Smarty 模板制作用户注册页面 光盘位置：光盘\MR\14\567	高级 趣味指数：★★★☆

实例说明

在本实例中，综合运用前面讲解的方法、技术，在模板页中直接嵌入 CSS 样式和 JavaScript 脚本，通过 foreach 语句输出验证码，通过 html_options 函数定义下拉列表框的值，并且增加 AJAX 技术实现用户名的无刷新验证，开发一个完整的 Smarty 用户注册功能模块，其运行结果如图 14.17 所示。



图 14.17　Smarty 模板中的用户注册

关键技术

通过 AJAX 技术实现无刷新验证整体上可以分为 3 个步骤：

（1）在模板页中，通过 script 标签调用指定的 JS 文件。创建 form 表单，添加表单元素，通过 onBlur 事件调用 JavaScript 脚本函数，完成对表单元素值的判断。

（2）编写 JavaScript 脚本文件，创建自定义函数，应用 AJAX 技术完成对表单元素值的验证。

（3）在模板页中，通过<div>标签输出验证结果。

其中应用到的 AJAX 技术如下：

（1）在使用 XMLHttpRequest 对象发送请求和处理响应之前首先需要初始化该对象，由于 XMLHttpRequest 不是一个 W3C 标准，所以对于不同的浏览器，初始化的方法也是不同的。

这里使用 IE 浏览器把 XMLHttpRequest 实例化为一个 ActiveX 对象，具体方法如下：

```
var http_request = new ActiveXObject("Msxml2.XMLHTTP");
```

或者

```
var http_request = new ActiveXObject("Microsoft.XMLHTTP");
```

在上面的语法中，Msxml2.XMLHTTP 和 Microsoft.XMLHTTP 是针对 IE 浏览器的不同版本而进行设置的。

（2）open()方法。open()方法用于设置进行异步请求目标的 URL、请求方法以及其他参数信息，语法如下：

```
open("method","URL"[,asyncFlag[,"userName"[, "password"]]])
```

在上面的语法中，method 用于指定请求的类型，一般为 GET 或 POST；URL 用于指定请求地址，可以使用

③ 发送 HTTP 请求，语法格式如下：

```
xmlobj.open(send_method,url,flag);
```

其中 xmlobj 是 XMLHttpRequest 的对象，send_method 是发送方法，可以是 GET 或者 POST，与表单中使用的方法相对应；url 是页面要调用的地址；flag 是一个标记，如果为 TRUE 则表示在等待被调用页面响应的时间内可以继续执行页面代码，反之为 FALSE。

设计过程

（1）定义 Smarty 的配置方法和 ADODB 连接、操作数据库的方法，有关内容请参考实例 565，这里不再赘述。

（2）创建 index.php 文件。首先设置页面的编码格式、载入配置文件，然后生成随机验证码，并且将随机验证码的值赋给模板变量，接着从数据库中读取数据，将读取的数据存储到数组中，并且将数组赋给模板变量，作为下拉列表框中的值，最后指定模板页。

（3）创建 index.html 模板页。首先创建 form 表单，添加表单元素，将用户的注册信息提交到 index_ok.php 文件中，其中，通过 html_options 自定义函数输出下拉列表框的值，通过 foreach 语句输出验证码的值，然后应用 literal 标签在页面中直接嵌入 CSS 样式，最后通过 script 标签载入 JavaScript 脚本文件，对用户提交的注册信息进行判断，其关键代码如下：

```
{literal}
<style type="text/css">
body,td,th {
 font-size: 12px;
}
</style>
{/literal}
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/check_form.js"></script>
<form id="form" name="form" method="post" action="index_ok.php" onsubmit="return check_form(); ">
用户名：<input name="user" type="text" id="user" onBlur="javascript:chkname()" size="20" />
<div id="user_check"><font color="#999999">请输入用户名</font></div>
词典选择：<select name="customer" id="customer">{html_options output=$cust_name values=$cust_id }</select>
验证码：<input type="hidden" id="checks" name="checks" value="{foreach key=key item=item from=$content} {$item} {/foreach}" />
<input name="check" type="text" id="check" size="8" />{foreach key=key item=item from=$content} {$item} {/foreach}
</form>
```

（4）创建 js 脚本文件夹。首先编写 createxmlhttp.js 文件，完成 AJAX 中 XMLHttpRequest 对象的初始化操作。然后编写 check_form.js 文件，定义 check_form()方法对表单中提交的数据进行验证；定义 chkname()方法调用 chkname.php 文件实现用户名的无刷新验证；定义 checkregemail(emails)方法验证邮箱地址的格式是否正确；定义 checkregtel(regtel)方法验证电话号码的格式是否正确。其中 chkname()方法的语法如下：

```
function chkname(){
 if(document.getElementById('user').value==""){                                              //判断用户名表单元素值是否为空
        document.getElementById('user_check').innerHTML="<font color=#FF0000>请输入用户名！</font>";   //如果为空则输出该内容
 }else{
        var user = document.getElementById('user').value;                                     //否则获取用户名
        var url = "chkname.php?user="+user;                                                   //定义 url
        xmlhttp.open("GET",url,true);                                                         //通过 GET 方法执行
        xmlhttp.onreadystatechange = function(){                                              //回调函数
        if(xmlhttp.readyState == 4){                                                          //判断返回值
                        var msg = xmlhttp.responseText;                                       //获取返回值
                        if(msg == '1'){                                                       //判断如果返回值为 1
        document.getElementById('user_check').innerHTML="<font color=#FF0000>用户名被占用！</font>";   //则说明用户名被占用
                                return false;
                        }else if(msg == '0'){                                                 //判断如果返回值为 0
                                document.getElementById('user_check').innerHTML="<font color=green>恭喜您，可以注册!</font>";    //用户名可用
                        }else{
                                document.getElementById('user_check').innerHTML="<font color=green>"+msg+"</font>";    //输出可用用户名
                        }
                }
        }
```

（1）在模板页中为不同的功能模块创建超链接，并通过超链接传递参数值。

（2）在动态 PHP 文件中，应用 switch 语句根据超链接中传递的参数值进行判断，从而包含不同的 PHP 脚本文件，并且将 PHP 脚本文件对应的模板文件名称通过 assign()方法赋给指定的模板变量。

（3）在模板页中，应用 Smarty 中的 include 函数加载模板变量传递的模板页。

这就是后台管理系统主页的设计方法，其中有关 switch 语句和 Smarty 模板 include 函数的详细讲解请读者参考本章实例 562，这里不再赘述。

设计过程

（1）创建 system 文件夹，编写 Smarty 编译、缓存和配置文件的存储目录 Smarty；编写 system.smarty.inc.php 文件，封装 Smarty 配置类、ADODB 连接和操作数据库类；编写 system.inc.php 文件，完成类的实例化操作。

（2）创建 index.html 文件，设计管理员登录页面；编写 login_ok.php 文件，完成管理员的登录操作。

（3）创建 main.php 文件。首先通过 header()函数设置页面的编码格式，初始化 SESSION 变量。然后根据 SESSION 变量判断当前用户是否具有访问权限。接着，如果具有访问权限，则应用 switch 语句根据超链接传递的参数值，完成在不同页面之间的跳转操作，即通过 include 函数包含不同的动态 PHP 文件，同时将动态 PHP 文件对应的模板文件名称赋给模板变量。最后指定模板页 main.html，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
session_start();                                                    //初始化 SESSION 变量
if($_SESSION['user']!="" and $_SESSION['pass']!=""){                //判断用户是否具有访问权限
require("system/system.inc.php");                                   //包含配置文件
switch ($_GET['caption']){                                          //完成在不同模块之间的跳转操作
 case "商品添加":
        include "sho_insert.php";                                   //包含 PHP 脚本文件
        $smarty->assign('admin_phtml','sho_insert.html');           //将 PHP 脚本文件对应的模板文件名称赋给模板变量
 break;
//省略了部分代码
 default:
        include "sho_update.php";
        $smarty->assign('admin_phtml','sho_update.html');
 break;
}
$smarty->assign("title","后台管理系统--".$_GET['caption']);         //定义模板变量
$smarty->assign("caption",$_GET['caption']);
$smarty->assign("type",$_GET['type']);
$smarty->assign("dates",date("Y 年 m 月 d 日"));                    //将当前时间值定义到模板变量中
$smarty->assign("user",$_SESSION['user']);                          //将当前登录用户的名称赋给模板变量
$smarty->display("main.html");                                      //指定模板页
}else{
        echo "<script>alert('您不具备访问权限！ '); window.location.href='index.html';</script>";
}
?>
```

（4）创建模板页 main.html。首先，输出 PHP 动态页中定义的模板变量值，包括页面的标题（$title）、当前时间（$dates）和当前页输出的模块类别（$type、$caption）。然后，通过 include 函数加载模板变量$admin_phtml 传递的模板页。最后，为后台管理系统中每个功能模块创建热点链接，并通过超链接的参数传递数据，同时应用 Smarty 模板中的 escape 方法对传递的参数值进行编码，其关键代码如下：

```
<!--载入模板文件-->
{include file=$admin_phtml}
<!--创建热点链接-->
<map name="Map" id="Map">
<area shape="rect" coords="29,41,88,62" href="main.php?caption={"商品添加"|escape:"url"}&amp;type={"商品管理"|escape:"url"}" />
<area shape="rect" coords="30,71,91,90" href="main.php?caption={"商品修改"|escape:"url"}&amp;type={"商品管理"|escape:"url"}" />
<area shape="rect" coords="31,99,91,118" href="main.php?caption={"商品删除"|escape:"url"}&amp;type={"商品管理"|escape:"url"}" />
</map>
```

（5）创建热点链接中链接的动态 PHP 文件和模板文件。由于篇幅所限，这里不对这部分内容进行详细讲解，读者可以参考本书光盘。

秘笈心法

心法领悟 568：Smarty 模板中的 URL 编码。

在本实例中，通过超链接传递参数值时，应用 Smarty 模板中的 escape 方法，对超链接传递的参数值进行编码，从而保护传递数据的安全。

实例 569	通过 section 循环输出数据 光盘位置：光盘\MR\14\569	高级 趣味指数：★★★☆

实例说明

在 PHP 动态页中，可以通过 while、do…while、for 和 foreach 语句实现数据的循环输出，而在 Smarty 中，也有属于自己的循环输出语句 foreach 和 section。在本实例中介绍通过 section 语句完成数据的循环输出，其运行结果如图 14.20 所示。



图 14.20　Smarty 模板下载页面

关键技术

section 循环语句用于比较复杂的数组，其语法如下：

```
{section name="sec_name" loop=$arr_name start=num step=num}
```

参数说明：

name：表示循环的名称。

loop：表示循环的数组。

start：表示循环的初始位置，如果 start=2，那么说明循环从 loop 数组的第 2 个元素开始。

step：表示步长，如果 step=2，那么循环一次后，数组的指针将向下移动两位，依此类推。

section 循环语句读取的是存储在模板变量中的数组元素，而这个数组元素值是在动态 PHP 文件中通过调用数据库操作类中的$admindb->ExecSQL 方法获取的。

究其根源就是应用 ADODB 类库中的 GetRows()方法获取的查询结果，有关其具体的设置可以参考封装的数据库操作类 AdminDB，该类存储于 system\system.smarty.inc.php 文件中。

设计过程

（1）本实例以实例 568 中开发的后台管理系统主页为基础，首先去除后台管理系统的登录功能，直接在 index.php 和 index.html 中编写后台管理系统的主页，在 switch 语句中，将 vip_look.php 和 vip_look.html 设置为默认值。index.php 的关键代码如下：



```
    </tr>
{/section}
```

在上述 section 语句中，将从数组的第 2 个元素开始循环，并且数组指针一次向下移动两位。

注意，在应用 section 语句时，一定要有{/section}结束标记。

实例 570 Smarty 模板中数据的分页显示

光盘位置：光盘\MR\14\570

高级

趣味指数：★★★★☆

实例说明

本实例在应用 section 语句循环输出数据库中数据的基础上，增加一个分页功能，实现对数据库中数据的分页显示，运行结果如图 14.21 所示。



图 14.21 Smarty 模板中数据的分页输出

关键技术

在 Smarty 中实现分页与在 PHP 文件中实现分页的原理是相同的，唯一的区别是，在 Smarty 中分页的处理操作与分页的显示是分离的。分页的处理操作存储在 sho_delete.php 文件中，其应用到的关键技术如下：

（1）mysql_query()函数执行查询、更改以及删除等操作，语法如下：

```
resource mysql_query ( string query [, resource link_identifier] )
```

参数 query 为字符串类型，指定传入的 SQL 指令；参数 link_identfier 为资源类型，指定传入的由 mysql_connect()函数或 mysql1_pconnect()函数返回的连接标识。如果省略该参数，则会使用最后一个打开的 MySQL 数据库连接。

（2）mysql_fetch_array()函数从数组结果集中获取数据，语法如下：

```
array mysql_fetch_array ( resource result [, int result_type] )
```

参数说明：

result：资源类型的参数，要传入的是由 mysql_query()函数返回的数据指针。

result_type：可选项，整数型参数，要传入的是 MYSQL_ASSOC、MYSQL_NUM 和 MYSQL_BOTH 这 3 种由 PHP 定义的常数之一，默认值是 MYSQL_BOTH。MYSQL_ASSOC 只得到关联索引（相当于 mysql_fetch_assoc()函数）；MYSQL_NUM 只得到数字索引（相当于 mysql_fetch_row()函数）；MYSQL_BOTH 将得到一个同时包含关联和数字索引的数组。

（3）array()函数定义一个数组，返回根据参数建立的数组。参数可以用“=>”运算符给出索引。array()函数是一个语言结构，用于字面上表示数组，不是常规的函数，语法如下：

```
array array ( [mixed ...])
```

参数 mixed 的语法为 key => value，多个参数间用逗号分开，分别定义了索引和值。

索引可以是字符串或数字。如果省略了索引，会自动产生从 0 开始的整数索引。如果索引是整数，则下一





图 14.22　Smarty+ADODB 完成数据的分页显示

■ 关键技术

应用 Smarty+ADODB 完成数据的分页显示，涵盖了 ADODB 连接、操作 MySQL 数据库以及 ADODB 中的分页技术，而 Smarty 依旧是将 ADODB 操作 MySQL 获取到的数据通过 assign()方法传递给模板变量，然后在模板页中应用 section 语句循环输出数据。

其中有关 ADODB 连接、操作 MySQL 数据库以及 ADODB 中分页技术的详细讲解，请读者参考本书第 13 章的内容，这里不再赘述。

在本实例中，将 Smarty 的配置方法以及 ADODB 连接和操作数据库的方法都存储在 system.smarty.inc.php 文件中；将 Smarty 配置类、数据库的连接和操作类的实例化存储在 system.inc.php 文件中；将 Smarty 的编译文件、配置文件和缓存文件的存储目录放置在 Smarty 文件夹下；而将 Smarty 类库和 ADODB5 类库存储在实例根目录的同级目录下。本实例的文件夹架构如图 14.23 所示。

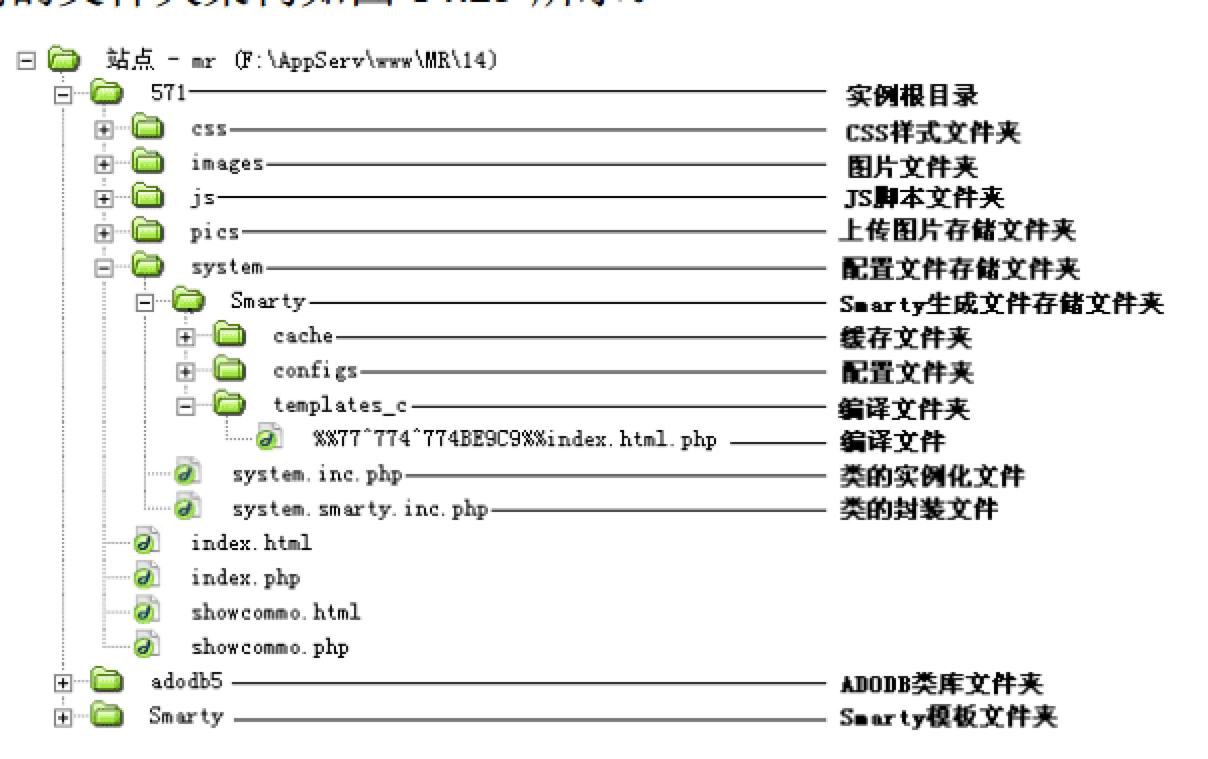


图 14.23　本实例的文件夹架构

■ 设计过程

（1）创建 system 文件夹，存储 ADODB 类库和 Smarty 模板文件。创建 system.smarty.inc.php 文件，定义

数据库连接类、数据库操作类和分页类。

数据库连接类 ConnDB，定义构造方法 ConnDB 为成员变量赋值；定义 GetConnId()方法，应用 NewADOConnection()函数连接 MySQL、mssql 或者 Access 数据库，设置数据库编码格式为 UTF-8，最后返回数据库连接对象；定义 CloseConnId()方法，关闭与数据库的连接。

数据库操作类 AdminDB，定义 ExecSQL()方法完成对数据库的添加、更新和删除操作。首先，通过 substr()函数截取 SQL 语句中前 6 个字符串，并将截取的字符串转换成小写。然后，通过 Execute()函数执行 SQL 语句。最后，根据截取的字符串判断 SQL 语句的类型，如果是 select 查询语句，则执行 GetRows()函数，如果查询结果为 0，则返回 FALSE，否则返回查询的数组；如果 SQL 语句为 update、insert 或者 delete 类型，执行成功则返回 TRUE，否则返回 FALSE。

分页类 SepPage，定义 ShowDate()方法完成从数据库中读取数据的操作，执行 PageExecute()函数实现分页功能，并且将返回值定义到数组中；定义 ShowPage()方法根据查询结果应用分页函数创建分页超链接，并且将返回结果定义到变量$str 中。

Smarty 配置类 SmartyProject，继承 Smarty 类库中 Smarty 类，设置本实例中 Smarty 缓存文件、配置文件、模板文件和编译文件的存储位置。

system.smarty.inc.php 文件的代码请参考本书光盘，由于篇幅所限这里不再介绍。

（2）创建 system.inc.php 文件，完成对数据库连接、操作、分页以及 Smarty 配置类的实例化操作，其代码如下：

```
<?php
require("system.smarty.inc.php");                                          //包含配置类文件
$smarty=new SmartyProject();                                               //实例化配置类
$connobj=new ConnDB("mysql","localhost","root","111","db_database14",false);     //实例化数据库连接类
$conn=$connobj->GetConnId();
$admindb=new AdminDB();                                                    //实例化数据库操作类
$seppage=new SepPage();                                                    //实例化分页类
?>
```

至此，有关 ADODB 和 Smarty 的安装和配置方法介绍完毕。

（3）创建 index.php 文件，调用分页类中的 ShowDate()方法，分页读取数据库中的数据，并且将返回值赋给模板变量；调用分页类中的 ShowPage()方法，完成分页超链接的输出，同样将返回值赋给模板变量，最后指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                                     //包含配置文件
$arraybbs=$seppage->ShowDate("select * from tb_commo where isnew = 1 order by id ",$conn,3,$_GET["page"]);  //调用分页类，实现分页功能
if(!$arraybbs){
    $smarty->assign("isbbs","F");
}else{
    $smarty->assign("isbbs","T");
$smarty->assign("showpage",$seppage->ShowPage("商品","个","","a1"));        //定义输出分页数据的模板变量 showpage
    $smarty->assign("arr",$arraybbs);
}
$smarty->assign('title','Smarty+Adodb 完成数据分页显示');
$smarty->display('index.html');
?>
```

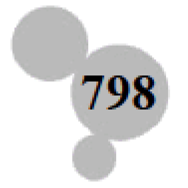
（4）创建 index.html 模板文件，应用 section 语句和模板变量完成数据的分页输出。

（5）创建 showcommo.html 和 showcommo.php 文件，用于输出指定产品的详细信息。

秘笈心法

心法领悟 571：在模板页中嵌入 PHP 脚本。

在 Smarty 模板页中也可以嵌入 PHP 脚本，应用的是 Smarty 中的 PHP 标签。PHP 标签中的语句是否被处理取决于$php_handling 的设置。通常情况下是不需要在模板页中嵌入 PHP 脚本的，但是不代表不可以使用。



实例 572	Smarty 模板中日期、时间的格式化输出 光盘位置：光盘\MR\14\572	高级 趣味指数：★★★☆

实例说明

在 PHP 脚本中，完成日期、时间的格式化输出最常用的是 date()函数，那么在 Smarty 模板中该如何完成日期、时间的输出呢？这就是本实例要讲解的内容，即通过 Smarty 模板中的 date_format 函数完成日期、时间的格式化输出，其运行结果如图 14.24 所示。



图 14.24　Smarty 模板中日期的格式化输出

关键技术

（1）Smarty 模板中的 date_format 函数用于格式化从 strftime()函数获得的时间和日期，其应用示例如下：

```
{$smarty.now|date_format:"%A, %B %e, %Y":$times}
```

参数说明：

$smarty.now：传递给 date_format 的数据。这个参数可以是在 PHP 动态页中定义的日期、时间模板变量，也可以使用 Smarty 中的保留变量$smarty.now 或者 Smarty 模板中的日期。

%A, %B %e, %Y：date_format 函数执行格式化操作时使用的格式。

$times：当传递给 date_format 的数据为空时，通过$times 设置 date_format 格式化的默认值。

（2）date_format 函数可以使用的转换格式很多，其常用的转换格式说明如下：

```
%a - 根据当地格式输出“星期”的缩写格式
%A - 根据当地格式输出“星期”的全称格式
%b - 根据当地格式输出“月”的缩写格式
%B - 根据当地格式输出“月”的全称格式
%Y - 根据当地格式输出“年”的全称格式
```

要了解 date_format 函数所有可以使用的转换格式，建议读者参考 Smarty 模板的参考手册。

设计过程

本实例继续使用实例 571 的内容，在 index.html 模板文件中插入 date_format 函数，输出系统的当前时间，其关键代码如下：

```
<td width="480" height="18" align="left">当前时间：  {$smarty.now|date_format:" %A   %B - %e - %Y"}</td>
<!--{$smarty.now|date_format}
{$smarty.now|date_format:"%A, %B %e, %Y":$times}
{$smarty.now|date_format:"%H:%M:%S"}
{$yesterday|date_format}
```

```
{$yesterday|date_format:"%A, %B %e, %Y"}
-->
```

本实例的其他内容可以参考实例 571，这里不再赘述。

秘笈心法

心法领悟 572：date_format 函数的妙用。

通过 date_format 函数不但可以输出系统的当前时间，如果配合 strtotime()函数还可以获取其他时间。例如，获取系统前一天的时间，其方法如下：

首先在 index.php 中应用 strtotime()函数获取前一天的时间戳，并且将时间戳赋给模板变量。然后在 index.html 模板页中应用 date_format 函数格式化输出前一天的时间，其关键代码如下：

```
//index.php 文件
$smarty->assign('yesterday', strtotime('-1 day'));
//index.html 文件
{$ yesterday |date_format:" %A    %B - %e - %Y"}
```

实例 573 Smarty 模板中的编码

光盘位置：光盘\MR\14\573

高级

趣味指数：★★★★☆

实例说明

在 PHP 脚本中应用 urlencode()函数对超链接中传递的参数进行编码，而在 Smarty 模板中提供 escape 函数对字符串进行编码。本实例将详细讲解该函数的应用，通过它对超链接中传递的参数进行编码，其运行结果如图 14.25 所示。

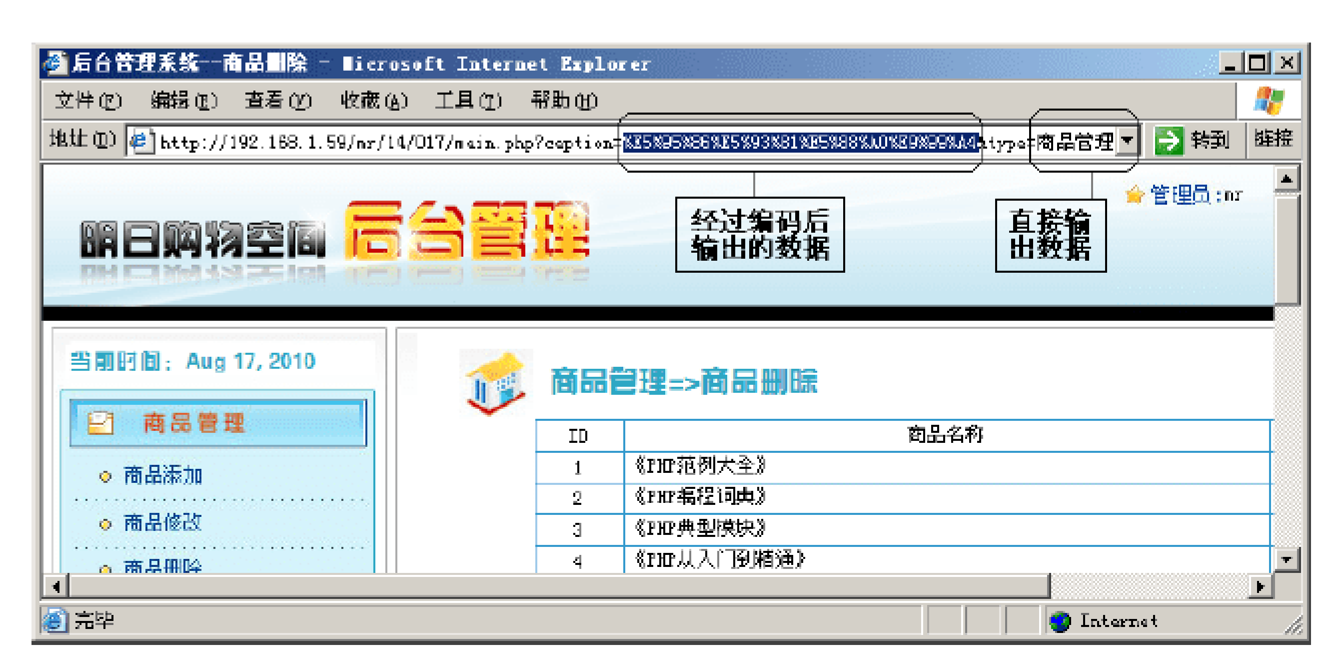


图 14.25 Smarty 模板中对超链接的参数值进行编码

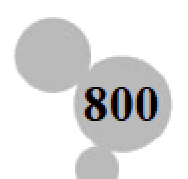
关键技术

escape 变量用于 HTML 转码、URL 转码，在没有转码的变量上转换单引号、十六进制转码或者 JavaScript 转码，默认是 HTML 转码，其基本的应用格式如下：

```
{$articleTitle|escape}
{$articleTitle|escape:"html"} {* escapes & " ' < > *}
{$articleTitle|escape:"htmlall"} {* escapes ALL html entities *}
{$articleTitle|escape:"url"}
{$articleTitle|escape:"quotes"}
<a href="mailto:{$EmailAddress|escape:"hex"}">{$EmailAddress|escape:"hexentity"}</a>
```

设计过程

（1）这里仍然应用实例 568 中的内容，为不同模块之间跳转的超链接传递的参数值编码，其具体的操作在 main.html 模板页中完成，关键代码如下：



例以实例 569 为基础，将分页输出的会员名称 mr 替换为明日科技，运行结果如图 14.26 所示。



图 14.26　Smarty 模板中正则表达式的运用

关键技术

regex_replace 为 Smarty 模板中的正则表达式，可对指定的字符串进行匹配，其包括两个参数：第一个是指定的正则表达式；第二个是指定的替换文本，其应用示例如下：

```
{$name|regex_replace:"/[mr]/":"<font color='#FF0000'>明日科技</font>"}
```

在这个示例中，将模板变量$name 中的 mr 替换为明日科技。

设计过程

这里以实例 570 的内容为基础，对会员浏览模块的内容进行重新编辑，分页输出会员浏览的数据，并且应用 regex_replace 变量将会员名称中的 mr 替换为明日科技。

（1）编辑 vip_look.php 文件，分页读取数据库中存储的会员数据，并且将分页读取返回的变量赋给模板变量，最终指定模板页，其关键代码如下：

```
<?php
include_once "conn/conn.php";                                    //连接数据库
require_once("system/system.inc.php");                           //调用指定的文件
//执行查询语句，从数据库中读取商品信息
$sql=mysql_query("select count(*) as total1 from tb_name ",$conn);
$info=mysql_fetch_array($sql);
$total1=$info[total1];                                           //统计数据库中数据总数
if(empty($_GET[pages])==true || is_numeric($_GET[pages])==false){ //判断变量 pages 是否为空
	$page1=1;                                                    //如果变量为空，则赋值为 1
}else{                                                           //如果不为空，则获取变量的值
	$page1=intval($_GET[pages]);
}
$pagesize1=3;                                                    //定义每页显示 3 条记录
if($total1<$pagesize1){                                          //判断如果数据库中的数据小于每页显示的记录数
	$pagecount1=1;                                               //则定义 pagecount 变量的值为 1
}else{
	if($total1%$pagesize1==0){
		$pagecount1=intval($total1/$pagesize1);                  //用总的记录数除以每页显示的记录数，获取共有几页
	}else{
		$pagecount1=intval($total1/$pagesize1)+1;
	}
}
//将要输出的数据赋给 assign 模板变量
$smarty->assign("total1",$total1);
$smarty->assign("pagesize1",$pagesize1);
$smarty->assign("page1",$page1);
$smarty->assign("pagecount1",$pagecount1);
$query=mysql_query("select * from tb_name order by id desc limit ".($page1-1)*$pagesize1.",$pagesize1",$conn );
$myrow=mysql_fetch_array($query);
$array=array();                                                  //定义一个空数组
```

这个功能，即 wordwrap。在本实例中将讲解这个变量的应用，并且通过它控制页面中输出字符串的行宽，其运行结果如图 14.28 所示。



图 14.28　每行显示 30 个字符串的运行效果

关键技术

wordwrap 指定段落的宽度（即控制一行显示多少个字符，如果超过这个字符数则换行），其默认值是 80 字符。该方法包含 3 个参数：

第 1 个参数设置每行显示的字符数。

第 2 个参数设置在约束点使用什么字符（默认值是换行符\n）。

第 3 个参数控制 Smarty 是截取到字符串的末尾，还是精确到指定长度的字符，默认情况是截取到字符串的默认值，如果设置第 3 个参数的值为 TRUE，那么就精确截取到指定长度的字符，其应用的示例如下：

```
{$articleTitle|wordwrap:30}
```

设置段落一行中包括 30 个字符，并使用\n 进行换行操作。

```
{$articleTitle|wordwrap:30:"<br>\n"}
```

设置段落一行中包括 30 个字符，并使用
\n 进行换行操作。

```
{$articleTitle|wordwrap:30:"\n":true}
```

设置段落一行中包括 30 个字符，并使用\n 进行换行操作，同时精确截取 30 个字符。

上述示例是对模板变量$articleTitle 中存储的数据进行操作。

设计过程

（1）创建 system 文件夹。首先定义 system.smarty.inc.php 文件，封装 Smarty 的配置方法。然后定义 system.inc.php 文件，对 Smarty 配置类进行实例化，并返回连接对象。最后创建 Smarty 文件夹，定义 Smarty 的编译文件、配置文件和缓存文件的存储目录。

（2）创建 index.php 文件。首先设置页面的编码格式、载入配置文件。然后通过 file_get_contents()函数读取存储在文本文件 content.txt 中的字符串，并应用 iconv()函数将字符串的 GB2312 编码转换为 UTF-8 编码，同时将转换后的文件存储到模板变量中，最终指定模板页，代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include("system/system.inc.php");                                        //包含配置文件
$smarty->assign('title','Smarty 模板中控制输出字符串的行宽');           //定义实例标题变量
$str = file_get_contents('files/content.txt');                           //读取文本文件中的数据
```

```
$smarty->assign('content',iconv("gb2312","utf-8",$str));          //将文本文件中的数据存储到模板变量中
$smarty->display('index.html');                                   //指定模板页
?>
```

（3）创建 index.html 模板页，通过模板变量输出文本文件中的字符串，并且应用 wordwrap 控制每行显示 30 个字符，应用
\n 完成换行操作，其关键代码如下：

```
{$content|wordwrap:30:"<br>\n"}
```

秘笈心法

心法领悟 576：Smarty 模板中换行符的转换。

在应用 wordwrap 控制段落中每行显示的字符时，默认使用\n 为换行符。但是使用\n 作为换行符，在页面中看不到字符串换行的效果，只有在查看源文件时才能看出字符串已经换行，所以在应用 wordwrap 控制段落中每行显示的字符时，应该设置第 2 个参数，指定
为换行符，这样在页面中就可以直接看到段落换行的效果。

在 Smarty 模板中，还可以通过 nl2br()函数将所有的换行符转换成</br>，其使用方法与 PHP 中的 nl2br()函数相同，而本实例中的字符串换行操作也可以使用如下方法来完成：

```
{$content|wordwrap:30|nl2br}
```

实例 577	register_object()方法注册模板对象 光盘位置：光盘\MR\14\577	高级 趣味指数：★★★★☆

实例说明

在 PHP 中，可以通过 function 创建自定义函数完成一些特殊的操作，同样在 Smarty 中可以通过 register_object()方法注册对象，在模板页中完成一些特殊的操作。在本实例中通过 register_object()方法注册对象，定义方法 moneyFormat()和 unHtml()，分别对留言本中的评分进行转义（保留两位小数）及对留言内容中的 HTML 标记进行转义，其运行结果如图 14.29 所示。



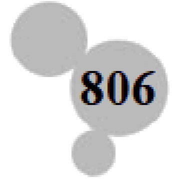
图 14.29 评分和 HTML 标记转义后输出

关键技术

（1）register_object()方法注册一个在模板中使用的对象，语法如下：

```
void register_object (string object_name, object $object, array allowed methods/properties, boolean format, array block methods)
```

register_object 注册对象的参数说明如表 14.5 所示。



定义数据库连接、操作、分页和 Smarty 的配置类，定义类的实例化文件 system.inc.php，完成各个类的实例化操作并返回操作对象。

（2）创建 index.php 文件，包含类的实例化操作文件；创建类，定义评分转义和 html 标记转义的方法，通过 register_object 注册模板对象；完成数据库中数据的分页读取，最终指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                                          //调用指定的文件
class Util {
        function moneyFormat($integral) {                                        //将数字转换成有两位小数的形式
                return str_replace ( ',', '', number_format ( $integral, 2 ) );
        }
        function unHtml($text) {                                                 //转换字符串中的 HTML 标签
                $str = htmlspecialchars ( $text );
                $str = ereg_replace ( '<br>', '\n', $str );
                $str = nl2br ( $str );
                $str = ereg_replace ( ' ', ' ', $str );
                return $str;
        }
}
$ntil=new Util();                                                                //实例化对象
$smarty->register_object("Util", $ntil,null,false);                              //注册模板对象
$array=$seppage->ShowDate("select * from tb_guestbook order by id desc",$conn,2,$_GET["page"]);          //调用分页类，实现分页
if(!$array){
        $smarty->assign("iscommo","F");                                          //判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
        $smarty->assign("iscommo","T");                                          //判断如果执行成功，则输出模板变量 iscommo 的值为 T
        $smarty->assign("showpage",$seppage->ShowPage("留言","条","","a1"));      //定义输出分页数据的模板变量 showpage
        $smarty->assign("array",$array);
}
$smarty->assign('title','register_object()方法注册模板对象');
$smarty->display('index.html');
?>
```

（3）创建 index.html 模板页，分页输出模板变量中传递的数据并且调用注册模板对象中的方法，对评分和 HTML 标记进行转换操作。同时创建一个 form 表单，用于提交留言信息，将数据提交到 index_ok.php 文件中。关键代码请参考关键技术中的内容。

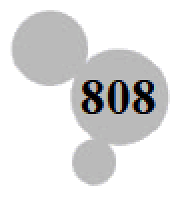
（4）创建 index_ok.php 文件，将表单中提交的留言信息存储到指定的数据表中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                             //设置文件编码格式
require_once("system/system.inc.php");                                          //调用指定的文件
$title=$_POST[title];                                                            //获取标题
$content=$_POST[content];                                                        //获取内容
$integral=$_POST[integral];
$createtime=date("Y-m-j H:i:s");                                                 //定义时间
$arr=$admindb->ExecSQL("insert into tb_guestbook(title,content,createtime,integral) values ('$title','$content','$createtime','$integral')",$conn);
if($arr){
        echo "<script>alert('留言发表成功!');window.location.href='index.php';</script>";
}else{
        echo "<script>alert('留言发表失败!');history.back();</script>";
}
?>
```

秘笈心法

心法领悟 577：模板对象的注销。

在 Smarty 模板中不但可以注册模板对象而且可以注销模板对象，注销模板对象应用的是 unregister_object() 方法。



```
$smarty->register_function("Util", "unHtml");                                   //注册模板函数
$array=$seppage->ShowDate("select * from tb_guestbook order by id desc",$conn,6,$_GET["page"]);     //调用分页类，实现分页
if(!$array){
        $smarty->assign("iscommo","F");                                         //判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
        $smarty->assign("iscommo","T");                                         //判断如果执行成功，则输出模板变量 iscommo 的值为 T
        $smarty->assign("showpage",$seppage->ShowPage("帖子","条","","a1"));     //定义输出分页数据的模板变量 showpage
     $smarty->assign("array",$array);
}
$smarty->assign('title','register_function()方法注册模板函数');
$smarty->display('index.html');

?>
```

（3）创建 index.html 模板页获取模板变量传递的数据，实现数据的分页输出，并且调用动态页中注册的模板函数实现对论坛内容的截取操作，其关键代码如下：

```
{section name=id loop=$array}
<tr>
<td colspan="4" bgcolor="#FFFFFF">{$array[id].id }  {Util text=$array[id].title }积分：{$array[id].integral}    
时间：{$array[id].createtime}<br>{Util text=$array[id].content}</td>
</tr>
{/section}
```

秘笈心法

心法领悟 578：字符串的截取。

通常使用 substr()函数对字符串进行截取，但是该函数在截取中文字符串时，很可能导致截取的中文字符串不完整出现乱码的情况。而本实例中应用的 mb_substr()函数同样可以实现字符串的截取功能，并且可以对中文字符串进行截取不会出现乱码的问题。

实例 579	Smarty 模板中 truncate 方法截取字符串 光盘位置：光盘\MR\14\579	高级 趣味指数：★★★★☆

实例说明

在 Smarty 中，提供了 truncate 方法对字符串进行截取，该方法可以截取到一个词的末尾。在本实例中，应用 truncate 方法对论坛中标题和内容进行截取，并且用省略号替换截取的内容，其运行结果如图 14.31 所示。



图 14.31　truncate 方法截取字符串

关键技术

truncate 方法从字符串开始处截取指定长度的字符，默认是 80 个字符。应用示例如下：

```
{$articleTitle|truncate}                    <!--截取默认长度的字符串-->
```

实例 580　开启网站注册页面的缓存　高级

光盘位置：光盘\MR\14\580　趣味指数：★★★☆

实例说明

缓存的合理应用应该以实际的开发需要为依据，对于那些需要经常更新的程序是否开启缓存，如果开启缓存，周期长短设置为多少，这些都必须根据程序的实际需求进行设置。在本实例中，开启网站注册页面的缓存，注册页面的运行结果如图 14.32 所示。



图 14.32　注册页面的运行结果

关键技术

缓存的设置既要考虑可以提高网站的访问速度，又要考虑缓存生命周期的合理性。不能偏执一方，应该采取中庸之道，达到最理想的效果。

例如，网站首页是不需要经常更新的内容，缓存的生存周期就可以设置长一些，而类似于论坛中帖子的数据，则可以不开启缓存，因为帖子的数据会不断地更新，如果使用缓存可能会导致浏览不到最新的数据。

（1）开启缓存。开启缓存的方法非常简单，只要将 Smarty 对象中$config 的值设置为 TRUE 即可，同时还要通过 Smarty 对象中的$cache_dir 属性指定缓存文件的存储位置，其操作代码如下：

```
$smarty->caching=true;                                          //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';            //定义缓存文件存储位置
```

（2）设置缓存生命周期。缓存创建成功后，必须为其设置一个生命周期，如果它一直不更新，那么就没有任何意义。设置缓存生命周期应用的是 Smarty 对象中的$cache_lifetime 属性，缓存时间以秒为单位，默认值是 3600 秒，其操作代码如下：

```
$smarty->caching=true;                                          //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';            //定义缓存文件存储位置
$smarty->cache_lifetime=3600                                    //设置缓存时间为 1 小时
```

（3）判断模板文件是否已经被缓存。如果页面已经被缓存，那么就可以直接调用缓存文件，而不再执行动态获取数据和输出的操作。为了避免在开启缓存后，再次执行动态获取数据和输出操作给服务器带来的压力，最佳的方法就是应用 Smarty 对象中的 is_cached()方法，判断指定的模板是否存在缓存，如果存在，则直接执行缓存中的文件，否则执行动态获取数据和输出的操作。操作代码如下：

```
$smarty->caching=true;                                          //开启缓存
```

实例 581　通过配置文件定义变量

光盘位置：光盘\MR\14\581

高级

趣味指数：★★★☆

实例说明

配置文件的应用，有利于设计者管理文件中的模板全局变量。例如，定义一个模板色彩变量，一般情况下如果想改变一个程序的外观色彩，必须更改每一个文件的颜色变量。如果有配置文件，色彩变量就可以保存在一个单独的文件中，只要改变配置文件就可以实现色彩的更新，这与在 Web 页面开发中应用的 CSS 样式非常相似。在本实例中，将讲解如何应用 Smarty 中的配置文件，并通过配置文件定义页面中 body 标签的样式，其运行结果如图 14.33 所示。



图 14.33　通过配置文件定义页面的样式

关键技术

（1）创建配置文件。

配置文件可以任意命名，其存储位置由 Smarty 对象的$config_dir 属性指定。如果存在不只在一个区域内使用的变量值，可以使用三引号（"""）将其完整地封装起来。在创建配置文件时，建议在程序运行前使用“#”加一些注释信息，这样有助于程序的阅读、更新。

在配置文件中既可以声明全局变量，也可以声明局部变量。如果声明局部变量，可以使用中括号“[]”括起来，在中括号之内声明的变量属于局部变量，而中括号之外声明的变量都是全局变量。中括号的使用不仅使配置文件中声明变量的模块变得清晰，而且可以在模板中选择加载中括号内的变量。

（2）加载配置文件。

加载配置文件应用 Smarty 的内建函数 config_load，其语法如下：

```
{config_load file="file_name " section="add_attribute" scope="" global=""}
```

参数说明：

file：指定包含的配置文件的名称。

section：附加属性，当配置文件中包含多个部分时应用，指定具体从哪一部分中取得变量。

scope：加载数据的作用域，取值必须为 local、parent 或 global。local 说明该变量的作用域为当前模板；parent 说明该变量的作用域为当前模板和当前模板的父模板（调用当前模板的模板）；global 说明该变量的作用域为所有模板。

global：说明加载的变量是否全局可见，等同于 scope=parent。

 **提示：**当指定 scope 属性时，可以设置 global 属性，但模板忽略该属性值，而以 scope 属性为准。

（3）引用配置文件中的变量。

配置文件加载成功后，就可以在模板中引用配置文件中声明的变量。引用配置文件应用的是“#”或者 Smarty 的保留变量$smarty.config，其应用示例如下：

```
{ config_load file="file_con.conf"}                    {* 加载配置文件 *}
{#title#}
<td height="228" colspan="2" align="left" valign="top" class="{$smarty.config.styles}">
```

设计过程

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录和配置文件目录；创建类文件 system.smarty.inc.php，定义 Smarty 的配置类，在该类中指定配置文件存储的目录，定义类的实例化文件 system.inc.php，完成 Smarty 类的实例化操作并返回操作对象。

（2）创建 index.php 文件，载入类的实例化文件；通过文件系统函数 file_get_contents()读取文本文件中的数据，并且将数据存储到模板变量中，最后指定模板页，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );          //设置文件编码格式
include("system/system.inc.php");                             //包含配置文件
$str = file_get_contents('files/content.txt');                //读取文本文件中的数据
$smarty->assign('content',iconv("gb2312","utf-8",$str));      //将文本文件中的数据存储到模板变量中
$smarty->display('index.html');                               //指定模板页
?>
```

（3）创建 index.html 模板页。首先通过 config_load()函数加载配置文件，然后应用“#”和$smarty.config 引用配置文件中的变量，最后通过模板变量输出文本文件中的数据，其关键代码如下：

```
{ config_load file="file_con.conf"}                    {* 加载配置文件 *}
<title>{#title#}</title>
<body leftmargin="{#leftmargin#}" topmargin="{#topmargin#}" marginwidth="{#marginwidth#}" marginheight="{#marginheight#}">
<td height="228" colspan="2" align="left" valign="top" class="{$smarty.config.styles}">{$content}</td>
```

秘笈心法

心法领悟 581：Smarty 模板的最佳配置方案。

在本章的实例中，应用面向对象技术将配置方法封装到类中，应用 Smary 模板时调用类的实例化文件即可。通过这种方法配置 Smarty 模板，不必再设置服务器的绝对路径和 Smarty 类库的绝对路径，非常简单、方便。

第15章

综合应用

- PHP 页面编码
- 网站公共文件设计
- 注册、登录
- 数据的输出
- 站内搜索
- 购物车
- 图书订单处理
- 后台管理系统登录
- 系统管理
- 图书类别管理
- 图书管理
- 用户管理
- 订单管理
- 通用信息管理

说明： 在文件编码下拉列表框中并没有 GB2312 编码，而 GBK 编码包含简体中文编码，所以这里选择 GBK 编码。

完成对页面文件的编码设置后，还需要使用 meta 标记设置页面编码，其设置方法如下：

```
<meta http-equiv="content-type" content="text/html; charset=gb2312" />
```

设计过程

（1）创建 index.php 页面，并设置页面文件的编码为 GB2312，其具体设置方法已经在关键技术中进行讲解，这里不再赘述。

（2）在 index.php 文件中通过 HTML 语言的 meta 标记设置网页编码为 GB2312 格式，具体实现代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>设计 GB2312 编码格式的网页</title>
        <meta http-equiv="content-type" content="text/html; charset=gb2312" />
        <link href="css/style.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <br/>
        <div style="width: 487px; text-align:center; height:221px; line-height:221px; font-size:16px; color:blue; background:url(img/bg.jpg)">
            这是一个 GB2312 编码格式的页面
        </div>
    </body>
</html>
```

秘笈心法

心法领悟 582：在 Dreamweaver 开发工具中设置文件的编码格式。

首先打开 Dreamweaver 开发工具，单击“编辑”按钮，选择“首选参数”命令，在弹出的“首选参数”对话框中通过“默认编码”下拉列表框设置创建文件的编码格式，如图 15.4 所示，设置完成后单击“确定”按钮。

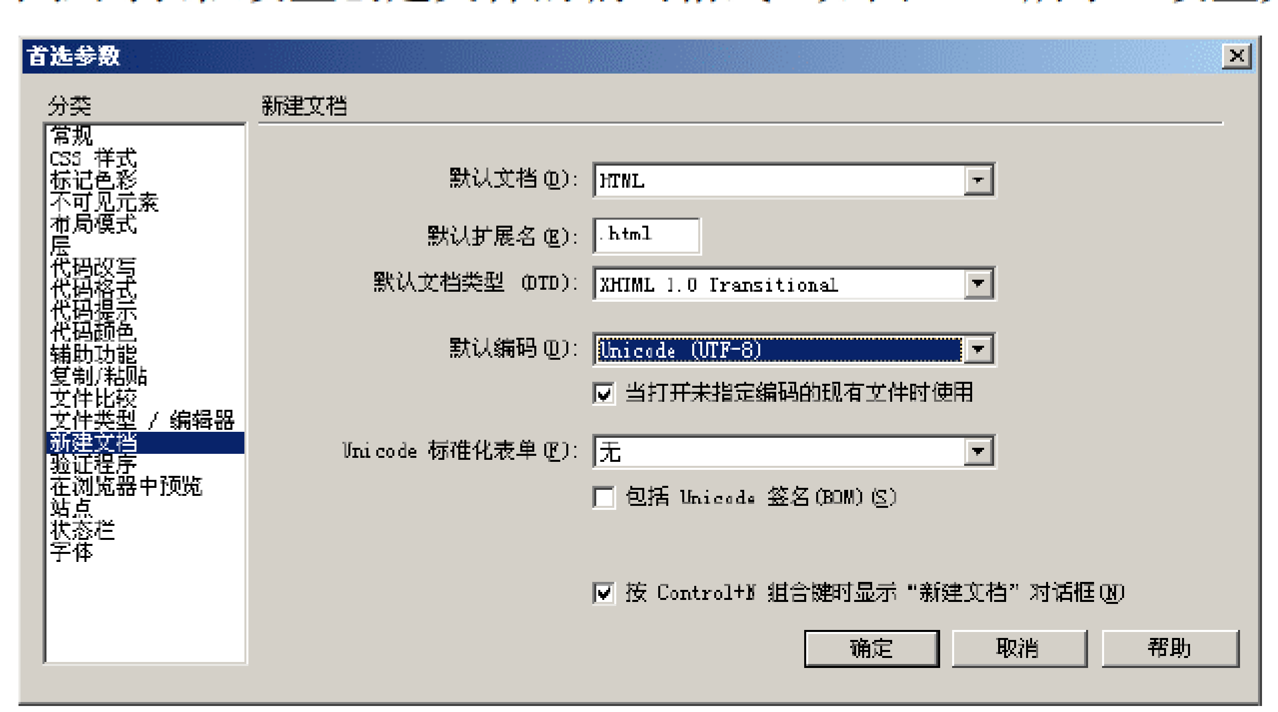


图 15.4　在 Dreamweaver 开发工具中设置文件的编码格式

实例 583　设计 GBK 编码格式的网页

光盘位置：光盘\MR\15\583

中级

趣味指数：★★★

实例说明

本实例主要讲解如何设置网页的页面编码格式为 GBK。运行本实例，如图 15.5 所示，在页面中打印出咖啡小屋程序所使用的页面编码。

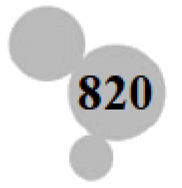




图 15.5　打印咖啡小屋的页面编码

关键技术

本实例的关键技术是设置网页的页面编码格式为 GBK，同样其设置方法也包括两个部分。首先设置网页文件的编码为 GBK 格式，其设置方法与实例 582 相同，这里不再赘述。然后使用 HTML 语言的 meta 标记设置网页的编码为 GBK，其设定代码如下：

```
<meta http-equiv="content-type" content="text/html; charset=gbk" />
```

设计过程

（1）创建 index.php 文件，并使用 Zend Studio 编辑器设置页面文件编码格式为 GBK。

（2）在 index.php 文件中使用 meta 标记设置网页编码格式为 GBK，并在页面中打印当前页面所使用的编码，其具体代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>设计 GBK 编码格式的网页</title>
        <meta http-equiv="content-type" content="text/html; charset=gbk" />
        <link href="css/style.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <br/>
        <div style="width: 498px; text-align:center; height:332px; line-height:221px; font-size:16px; color:#FF3300; background:url(img/bg.jpg)">
            这是一个 GBK 编码格式的页面
        </div>
    </body>
</html>
```

秘笈心法

心法领悟 583：GB2312 和 GBK。

GB2312 和 GBK 是中国政府颁布的标准汉字编码。GB2312 编码中放置了 6763 个常用汉字，并且首次在中文操作系统 CCDOS 中使用；GBK 编码在 GB2312 编码的基础上，增加了我国台湾 BIG5 编码中的 15000 多个汉字的字符。

实例 584	设计 UTF-8 编码格式的网页 光盘位置：光盘\MR\15\584	中级 趣味指数：★★★

实例说明

本实例主要讲解如何设置网页页面编码格式为 UTF-8。运行本实例，如图 15.6 所示，将在页面中打印出企

业邮局收发系统所使用的页面编码。



图 15.6　打印企业邮局收发系统页面编码

关键技术

UTF 是国际字符集转换格式（Unicode/UCS Transformation Format）的缩写，是 FCS 的实际表达式，按其基本长度所用位数分为 UTF-8/16/32 这 3 种形式，其中 UTF-8 保持字符数字一个字节，其他的用不定长编码到最多 6 个字节，支持到 31 位编码。UTF 中最常用的就是 UTF-8 编码格式，UTF-8 使用 8 位字节编码，通常使用 1~3 位字节来表示一个字符，字符的编码长度为变长。

UTF-8 字符集不仅包括欧美字符，也包括中国、日本和韩国的全角字符。因此，在网页中使用 UTF-8 字符集可以使用户在不安装中文编码的情况下浏览到大部分中文字符。

本实例设置页面为 UTF-8 编码，首先需要使用编辑工具设置页面文件编码格式为 UTF-8，然后使用 HTML 的 meta 标记设置网页编码格式为 UTF-8，其设置方法如下：

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

设计过程

（1）新建 index.php 文件，并使用 Zend Studio 编辑器设置页面文件编码格式为 UTF-8。

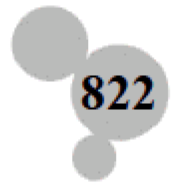
（2）在 HTML 语言的 meta 标记中设定当前页面所使用的编码格式为 UTF-8，其具体代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>设计 UTF-8 编码格式的网页</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <link href="css/style.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <br/>
        <div style="width: 978px; text-align:center; height:591px; line-height:591px; font-size:16px; color:#09600C; background:url(img/bg.jpg)">
            这是一个 UTF-8 编码格式的页面
        </div>
    </body>
</html>
```

秘笈心法

心法领悟 584：Web 程序开发中页面编码格式的选择。

为了便于程序的国际化及使程序支持更多语言版本的操作系统，在项目开发中建议将页面设置为 UTF-8 编码格式。



实例 585　PHP 的国际化

光盘位置：光盘\MR\15\585

中级

趣味指数：★★★

实例说明

信息全球化的高速发展，使企业的市场也在逐步扩大，并逐渐走向世界。企业向世界展示自己，必不可少地要应用到互联网，在应用互联网的过程中必须要提供多国语言的支持。只有这样才能让更多的人了解企业。所以在开发程序的过程中，必须要提供对多国语言的支持。本实例以一个简单的企业内部管理系统为例，介绍如何通过 PHP 语言开发提供多国语言支持的程序。在实例中，实现文章的添加和浏览功能，并且提供对中文和英文两种语言的支持。中文文章添加页面如图 15.7 所示，英文文章添加页面如图 15.8 所示。

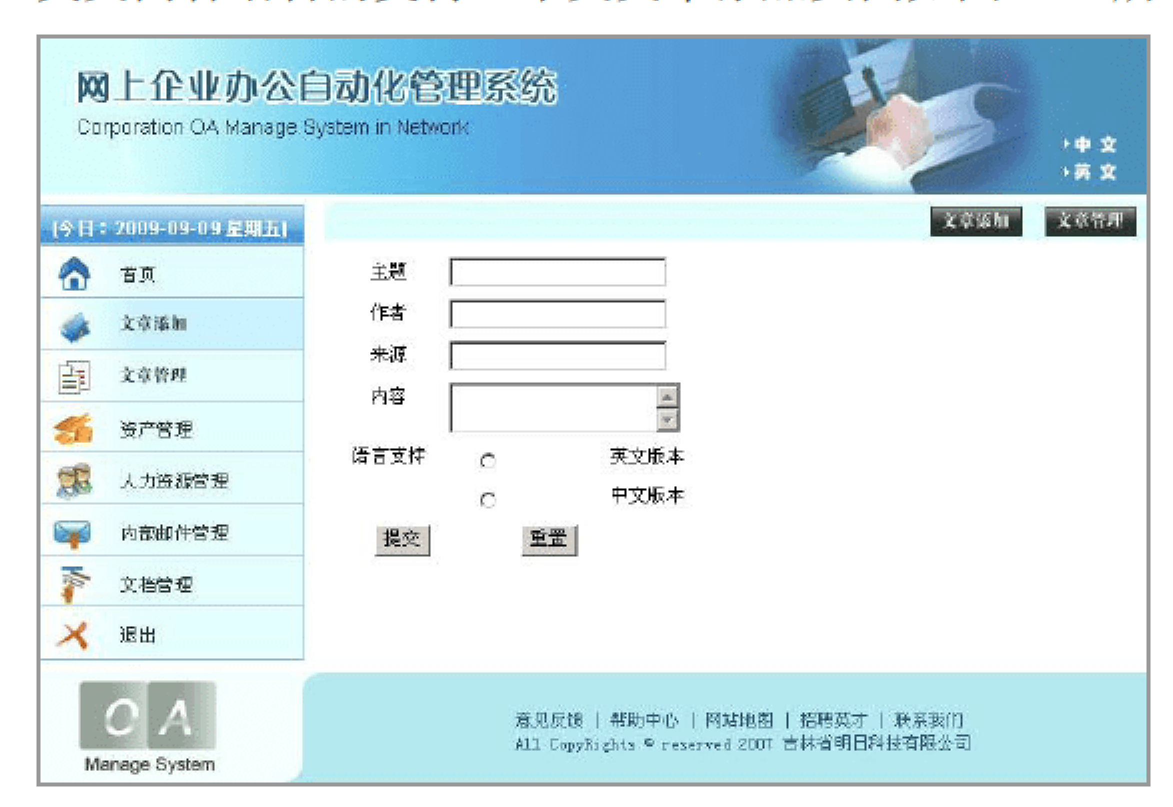



图 15.7　中文文章添加页面



图 15.8　英文文章添加页面

关键技术

本实例中应用到的关键技术如下：

（1）网页编码的设置。在网页中，字符编码通过<head>标签中<meta>标签的 charset 属性来设置，代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>设置网页编码格式</title>
</head>
<body>通过 meta 标签的 charset 属性设置字符编码格式</body>
</html
```

（2）获取客户端语言选项，根据语言选项的值调用不同的语言配置文件，代码如下：

```
<?php
session_start();
$lang=$_SERVER['HTTP_ACCEPT_LANGUAGE'];          //获取语言选项
$lang=explode(';',$lang);                        //以“;”对字符串进行分隔
$lang=$lang[0];                                  //为数组赋值
$lang=explode(',',$lang);                        //以“,”对字符串进行分隔
$lang=$lang[0];                                  //为数组赋值
switch($lang){                                   //根据获取到的语言选项的值为变量赋值
 case "en-us":
        $lang="en";
        break;
 case "zh-cn":
        $lang="ch";
        break;
 default :
```

```
        $lang="en";
        break;
}
if(isset($_SESSION['lang'])){
        $lang=$_SESSION['lang'];                    //为 SESSION 变量赋值
}
include "lang/lang_$lang.php";                      //调用不同的语言配置文件
?>
```

其中，$_SERVER['HTTP_ACCEPT_LANGUAGE']为服务器变量，用于获取客户端浏览器的语言设置。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与数据库的连接，其代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or die('连接失败:' . mysql_error());
mysql_select_db("db_database15",$conn) or die ('数据库选择失败:' . mysql_error());
mysql_query("set names gb2312");
?>
```

（2）创建 lang 文件夹，编写 lang_ch.php 和 lang_en.php 文件，分别定义中文语言配置和英文语言配置文件，代码请参考附送光盘中的内容。

（3）创建 config.php 文件，获取浏览器中语言选项的值，根据语言选项的值调用不同的语言配置文件。其完整代码可以参考关键技术中的内容。

（4）创建 setlang.php 文件，根据超链接中传递的参数为 SESSION 变量赋值，并且调用不同的操作页面，其代码如下：

```
<?php
session_start();
switch($_GET['lang']){                                  //获取超链接传递的语言选项的值
 case "en":
        $_SESSION['lang']="en";                         //根据超链接的值为 SESSION 变量赋值
        break;
 case "ch":
        $_SESSION['lang']="ch";
        break;
 default :
        $_SESSION['lang']="en";
        break;
}
if($_GET['link']=="addnew"){                            //根据超链接的值跳转到不同操作页面
        header("Location:addnew.php");
}elseif ($_GET['link']=="show"){
        header("Location:show.php");
}else{
        header("Location:index.php");
}
?>
```

（5）创建 index.php 文件，根据选择的语言不同输出不同的页面和内容。首先通过 require_once 语句调用不同的包含文件，完成数据库连接和语言配置。然后执行查询操作，从数据表中读取指定的数据信息。最后在页面中输出数据，其关键代码如下：

```
<?php
require_once('conn/conn.php');                          //连接数据库
require_once('config.php');                             //调用语言配置文件
$query_rs="select * from tb_articles where langver='$rs_filter' order by id desc";   //定义查询语句
$rs=mysql_query($query_rs,$conn) or die(mysql_error()); //执行查询操作
$row_rs=mysql_fetch_assoc($rs);                         //获取查询结果的记录数
?>
```

（6）创建 add_data.php 文件，添加 form 表单用于提交数据并完成对表单中数据处理的操作。首先通过 require_once 语句调用不同的包含文件，完成数据库连接和语言配置。然后定义方法 GetSQLValueString()完成对数据格式的转换。接着获取表单中提交的数据，并应用 sprintf()函数和 GetSQLValueString()方法完成对数据格式的转换将数据添加到数据表中。最后设计文章添加页面，创建 form 表单，完成数据的提交操作，其关键代码如下：





图 15.9　明日科技书店成功连接 MySQL 数据库的首页运行效果

关键技术

本实例的关键是设计一个可以对多种数据库进行连接的数据库连接类。下面对数据库连接类的制作方法进行详细讲解：

（1）首先定义数据库连接类的属性，关键代码如下：

```
var $dbType;                                    //数据库类型标识
var $host;                                      //数据库服务器地址
var $userName;                                  //数据库服务器用户名
var $password;                                  //数据库用户密码
var $dbName;                                    //要连接的数据库名
var $isDebug;                                   //是否显示调试信息
var $connID
```

说明：这里为了兼容 PHP 5.0 以下的版本，所有属性都定义成 var 类型。

（2）定义数据库连接类的构造方法，实现对类内属性的初始化，其关键代码如下：

```
function ConnDB ($dbType = 'mysql', $host, $userName, $password, $dbName, $isDebug = false){
    $this->dbType = $dbType;                    //为数据库类型赋初值
    $this->host = $host;                        //为服务器地址赋初值
    $this->userName = $userName;                //为用户名赋初值
    $this->password = $password;                //为密码赋初值
    $this->dbName = $dbName;                    //为数据库名赋初值
    $this->isDebug = $isDebug;                  //为调试信息赋初值
}
```

说明：上述代码中的“->”是 PHP 面向对象编程方式中，实例化的对象引用类中方法或属性的标识符。

（3）通过 ConnDB()构造方法对类中属性初始化后，就可以实现与数据库的连接。具体连接过程通过方法 getConnID()实现，在该方法中根据属性$dbType 的值来判断要连接数据库的类型，并根据不同的数据库类型选择相应的数据库连接驱动最终返回连接对象，其关键代码如下：

```
function getConnID (){
    require_once 'library/adodb/adodb.inc.php';                  //导入 ADODB 类库
    $this->connID = NewADOConnection($this->dbType);             //生成数据库连接对象
    if ($this->dbType == 'mysql' || $this->dbType == 'mssql') {  //如果为 MySQL 数据库或者 SQL Server 数据库
        $this->connID->Connect($this->host, $this->userName, $this->password, $this->dbName);  //通过 Connect()方法建立与数据库的连接
        if ($this->dbType == 'mysql') {
            $this->connID->Execute('set names gb2312');          //设置数据库编码
        }
    } elseif ($this->dbType == 'access') {                       //如果为 Access 数据库
```



图 15.10　图书信息简介

关键技术

在数据库管理类 AdminDB 中，只有一个名为 executeSQL()的方法，通过该方法可以实现对数据库的所有查询和维护操作，其关键代码如下：

```
class AdminDB{
    function executeSQL ($sql, $connID){
     $sqlType = strtolower(substr(trim($sql), 0, 6));           //提取 SQL 语句的类型
        $rs = $connID->Execute($sql);                           //执行 SQL 语句
        if ($sqlType == 'select') {                             //如果是 select 查询
         $arrayData = $rs->GetRows();                           //返回查询记录集
            if (count($arrayData) == 0 || $rs == false) {       //如果没查询到或发生错误
                return false;                                   //返回 FALSE
            } else {                                            //否则
                return $arrayData;                              //返回记录集
            }
        } elseif ($sqlType == 'insert' || $sqlType == 'update' || $sqlType == 'delete') {
            //如果执行插入、更新或删除语句
            return $rs;                                         //返回语句执行状态，即成功返回 TRUE，失败返回 FALSE
        } else {
            return false;                                       //如果不是上述查询，则返回 FALSE
        }
    }
}
```

上述数据库管理类中的 executeSQL()方法包含$sql 和$connID 两个参数，分别为要执行的 SQL 语句和数据库连接对象。在函数体内，首先将传入的 SQL 语句转换成小写，然后提取 SQL 语句的前 6 个字符，同时通过数据库连接对象的 Execute()方法执行 SQL 语句，并根据提取的 SQL 语句的前 6 个字符判断 SQL 语句的类型，从而返回查询结果。

设计过程

（1）建立数据库管理类 AdminDB，该类的创建过程已经在本实例的关键技术中进行了详细介绍，这里不再赘述，然后使用 new 关键字对该类进行实例化，实现代码如下：

```
$adminDB = new AdminDB();
```

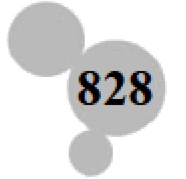
（2）使用 AdminDB 类实例的对象调用该类中的 executeSQL()方法执行查询，并将查询结果赋给视图变量，实现该过程的关键代码如下：

```
$bookinfo = $adminDB->executeSQL("select tb_bookinfo.id as bid, bookname, bookids, page, isbn, zs, bookcc, bc, about, directory, writer, pubname, pubtime, bookimg, typename, oldprice, newprice from tb_bookinfo, tb_pub, tb_smalltype where tb_bookinfo.pubid = tb_pub.id and tb_bookinfo.smalltypeid = tb_smalltype.id and tb_bookinfo.id = '" . $bid . "'", $connID);        //查询图书信息
$smarty->assign('bookinfo', $bookinfo);                       //将图书信息赋给视图变量
```

（3）在 bookinfo.phtml 视图文件中，根据传递的视图变量的值打印图书的各项信息，其具体实现代码请详见本书附带光盘。

秘笈心法

心法领悟 587：根据 select、update 和 delete 关键字的字符串长度完成 SQL 语句类型的判断。



```
            $arrayPageInfo['countPage'] = $countPage;                               //总页数
            $arrayPageInfo['first'] = 1;                                            //第一页的页码
            if ($page > 1) {
                $arrayPageInfo['previous'] = $rs->AbsolutePage() - 1;               //前一页的页码
            } else {
                $arrayPageInfo['previous'] = 1;
            }
            if ($page < $countPage) {
                $arrayPageInfo['next'] = $rs->AbsolutePage() + 1;                   //后一页的页码
            } else {
                $arrayPageInfo['next'] = $countPage;
            }
            $arrayPageInfo['last'] = $countPage;                                    //最后一页的页码
        }
        return $arrayPageInfo;
    }
}
```

上述分页类实现的原理是：设法使用 ADODB 数据抽象层获取指定页码中所显示的数据，同时将首页、上一页、下一页、尾页、总页码和总记录数都作为数组元素，然后将该数组作为分页类中实现分页方法的返回值返回，在需要分页的模块，只需要通过分页类获取该返回数组就可以获得所有分页所需信息。

在实现上述 PageDB 分页类时，在类体内只定义一个 pageData()分页方法，在该方法内部，通过 ADODB 数据抽象层根据分页方法的$pageSize 和$page 参数获取页面显示的数据和分页导航参数，并保存到$arrayPageInfo 数组中返回。

设计过程

（1）建立数据库分页类 PageDB，并在该类中建立 pageData()方法，该类的创建过程已经在关键技术中给出，这里不再赘述。

（2）在 listsepbook.php 文件中，包含网站头文件，根据超链接传递的参数值为变量赋值，调用数据库管理类中的方法执行查询语句，并且将返回的查询结果赋给模板变量。定义分页变量，调用分页类中的方法完成数据的分页输出，同时将分页方法返回的数据赋给模板变量，最后指定模板页，其关键代码如下：

```
require_once 'header.php';                                                          //包含 header.php 文件
if ($_GET['t'] == 'new') {                                                          //展示新书
    $nowtype = '新书推荐';
    $andwhere = "isnew = 1";
} elseif ($_GET['t'] == 'sepprice') {                                               //展示特价图书
    $nowtype = '特价图书';
    $andwhere = "issepprice = 1";
} elseif ($_GET['t'] == 'hotsell') {
    $nowtype = '热卖图书';                                                          //展示热卖图书
    $andwhere = "ishotsell = 1";
} elseif ($_GET['t'] == 'term') {
    $nowtype = '期书';
    $andwhere = "isterm = 1";
} elseif ($_GET['t'] == 'mrbook') {                                                 //展示明日图书
    $nowtype = '明日图书推荐';
    $andwhere = "ismrbooktj = 1";
}
$smarty->assign('gt', $_GET['t']);                                                  //为模板变量赋值
$smarty->assign('nowtype', $nowtype);
$system = $adminDB->executeSQL("select bookimgurl, readurl from tb_system where mark = 1", $connID);   //执行查询语句
$smarty->assign('system', $system);                                                 //将查询结果赋给模板变量
if (! isset($_GET['page']) || $_GET['page'] == '') {                                //判断分页变量的值
    $page = '1';                                                                    //设置分页变量的值
} else {
    $page = $_GET['page'];
}
$sql = "select tb_bookinfo.id, tb_bookinfo.bookimg, tb_bookinfo.about, tb_bookinfo.browsertime, tb_bookinfo.oldprice, tb_bookinfo.newprice,
tb_bookinfo.bookname, tb_bookinfo.writer, tb_bookinfo.pubtime, tb_bookinfo.addtime, tb_bookinfo.ishave, tb_pub.pubname from tb_bookinfo, tb_pub
where tb_bookinfo.pubid = tb_pub.id and " . $andwhere . " order by tb_bookinfo.addtime desc";              //定义查询语句
$bookinfos = $pageDB->pageData($sql, $connID, 10, $page);                           //调用分页类中的方法，执行查询操作
```

上述代码首先通过函数 require_once()包含 Smarty 类库中的 Smarty.class.php 文件，从而将 Smarty 类库导入到工程中，然后定义 SmartyConfig 类，使之继承自 Smarty 类，在该类的构造方法中，首先通过$this 关键字调用父类的构造方法，实现对 Smarty 模板引擎的初始化，然后使用$this 关键字分别调用 Smarty 模板中的属性对模板路径进行配置。

设计过程

（1）下载 Smarty 模板类库，将其中的 libs 文件夹复制到项目的 library\smarty 文件夹下。

（2）在 library\smarty 文件夹下创建 cache_dir、compile_dir 和 config_dir 这 3 个文件夹，用于存储 Smarty 的缓存文件、编译文件和配置文件。

（3）在 library 文件夹下创建 SmartyConfig.php 文件，编写 Smarty 模板引擎配置类 SmartyConfig，其代码可以参考关键技术中的内容。

秘笈心法

心法领悟 589：读者需自行下载 Smarty 模板类库。

由于 Smarty 模板类库属于第三方组件，所以在光盘源程序中没有提供，需要读者自行下载，并且将下载文件中的 libs 文件夹复制到项目的 library\smarty 文件夹下。

实例 590	字符串处理类 光盘位置：光盘\MR\15\590	中级 趣味指数：★★★

实例说明

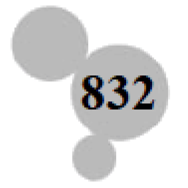
在制作明日科技书店网站时，很多模块都需要实现对中文字符串的截取、HTML 标记的原样输出、CKEditor 在线编辑器的调用等功能。为了提高代码的重用率，可将上述功能封装成字符串处理类。明日网上书店新书推荐中图书名称的输出、图书价格的输出，以及新闻公告中新闻标题的输出都应用了字符串类中的方法，其运行效果如图 15.13 所示。



图 15.13　字符串类中方法的应用

关键技术

在字符串处理类 Util 中定义多个方法，分别实现对货币格式化、对 HTML 标记进行转义输出、对中文进行截取、防止出现乱码等。Util 类中的方法如表 15.1 所示。



```
function msubstr ($str, $start, $len){
    $strlen = $start + $len;                                    //获取截取字符长度
    for ($i = 0; $i < $strlen; $i ++) {                         //遍历所有字符
        if (ord(substr($str, $i, 1)) > 0xa0) {                  //判断是否为中文字符
            $tmpstr .= substr($str, $i, 2);                     //如果是中文则截取 2 个字节
            $i ++;
        } else
            $tmpstr .= substr($str, $i, 1);                     //不是中文则截取 1 个字节
    }
    return $tmpstr;                                             //返回结果
}
```

（7）使用 HTML 语言提供的文件域对文本进行编辑。实现文本特效较复杂，而且不能达到“所见即所得”的效果，为了解决上述问题，在进行 Web 项目开发时，一般使用 CKEditor 在线编辑器作为表单的文本编辑域。在制作明日科技书店时会在很多模块使用线编辑器，所以在开发时将调用 CKEditor 编辑器的代码封装在 Util 字符串处理类中，具体代码如下：

```
function editor ($name, $value, $width = '100%', $height = '200'){
    require_once 'library/fckeditor/fckeditor.php';                 //导入 CKEditorr 类库
    $arrayIni = $this->arrayIni;                                    //获取配置文件内容
    $oFCKeditor = new FCKeditor($name);                             //实例 CKEditor
    $oFCKeditor->BasePath = $arrayIni['baseUrl'] . '/library/fckeditor/';  //设置 CKEditor 路径
    $oFCKeditor->Width = $width;                                    //编辑器的宽度
    $oFCKeditor->Height = $height;                                  //编辑器的高度
    $oFCKeditor->ToolbarSet = 'Default';                            //设置工具栏样式
    $oFCKeditor->Value = $value;                                    //初始值
    $oFCKeditor->Create();                                          //创建编辑器
}
```

（8）完成 Util 类的封装，在 lzh.inc.php 文件中应用 Smarty 中的 register_object()方法将其注册为模板对象，在模板页中通过模板对象调用类中方法实现对字符串的操作。

在 lzh.inc.php 文件中不只实现注册模板对象的功能，所有类的实例化操作都在该文件中完成，并返回实例化对象，包括数据库连接类、数据库操作类、分页类和 Smarty 配置类，其代码如下：

```
<?php
session_start();                                                //初始化 SESSION 变量
$arrayIni = parse_ini_file('config/lzhConfig.ini');             //包含配置文件
require_once 'library/ConnDB.php';                              //载入数据库连接类文件
require_once 'library/AdminDB.php';                             //载入数据库操作类文件
require_once 'library/PageDB.php';                              //载入分页类文件
require_once 'library/SmartyConfig.php';                        //载入 Smarty 配置类文件
require_once 'library/Cart.php';                                //载入购物车类文件
require_once 'library/Util.php';                                //载入字符串处理类文件
$connDB = new ConnDB($arrayIni['dbType'] ,$arrayIni['host'] ,$arrayIni['userName'] ,$arrayIni['password'] ,$arrayIni['dbName'] ,$arrayIni['isDebug']);
$connID = $connDB->getConnID();                                 //执行与数据库的连接
$adminDB = new AdminDB();                                       //数据库操作类实例化
$pageDB = new PageDB();                                         //分页类实例化
$smarty = new SmartyConfig();                                   //Smarty 配置类实例化
$util =   new Util();                                           //字符串操作类实例化
$smarty->register_object('util', $util, null, false);          //注册模板对象
```

（9）在 index.php 文件中，调用数据库操作类中的方法，查询出数据库中的数据，并且将查询结果赋给模板变量，最终指定模板页。

（10）在模板页 views\index.phtml 中，应用 section 语句循环输出模板变量中的数据，并且通过注册对象调用字符串类中的方法对输出的图书标题、图书价格和公告标题等进行格式化操作，其关键代码如下：

```
{section name=rID loop=$reads}
<div style="width:198px; padding-bottom:7px">
 <div style="width:100%; height:90px">
        <div style="width:85px; height:100px; float:left">
                <a href="{util->baseUrl}/bookinfo-{$reads[rID].bookid}.html"><img src="{$system[0].bookimgurl}/{$reads[rID].bookimg}"
width="70" height="90"   style="border:1px solid #123456" /></a>
           </div>
           <div style="width:112px; height:100px; float:right; text-align:left">
                <li style="display:inline; height:35px"><a href="{util->baseUrl}/bookinfo-{$reads[rID].bookid}.html" class="a6">•{util->unHtml
p=$reads[rID].bookname}</a></li>
```

```
<?php
require_once 'lzh.inc.php';                                          //包含类的实例化文件
$requestPageName = basename ( $_SERVER ['PHP_SELF'] );               //获取当前文件的名称
switch ($requestPageName) {                                          //根据当前文件的名称定义页面的标题
 case 'index.php' :
        $title = '明日网上书店 - 国内信誉一流的 IT 类图书网上专营店，最全面、最资深的网上书店！';
        break;
 case 'bookinfo.php':
        $topbookinfo = $adminDB->executeSQL("select bookname from tb_bookinfo where id='".$_GET['bid']."'", $connID);
        $title = '明日网上书店 - '.$topbookinfo[0]['bookname'];
        break;
 case 'listbook.php':
        $topsmalltypeinfo = $adminDB->executeSQL("select typename from tb_smalltype where id='".$_GET['stid']."'", $connID);
        $title = '明日网上书店 - ' . $topsmalltypeinfo[0]['typename'] . ' 类图书';
        break;
 case 'listsepbook.php':
        $topt = $_GET['t'];
        if($topt == 'new'){
               $title = '明日网上书店 - 新书推荐';
        }
        //省略了部分代码
}
$smarty->assign('title', $title);                                    //将标题的内容赋给模板变量
if(isset($_SESSION['unc']) && $_SESSION['unc']!=''){                 //判断当前用户是否为空
     $smarty->assign('isLogin', 'T');                                //如果当前用户不为空，则为模板变量赋值 T
     $smarty->assign('unc', $_SESSION['unc']);                       //并且将当前用户名称赋给模板变量
}
function getSidByBid($params){                                       //创建自定义函数
     extract($params);
     global $adminDB, $connID;                                       //定义变量范围
     $stype = $adminDB->executeSQL("select id from tb_smalltype where bigtypeid='".$bid."' order by addtime limit 0, 1", $connID);
     return $stype[0]['id'];                                         //获取 ID 值
}
$smarty->register_function('getSidByBid', 'getSidByBid');            //注册模板函数
$bigtypeDhs = $adminDB->executeSQL("select id, typename from tb_bigtype order by addtime", $connID);    //执行查询操作
$smarty->assign('bigtypeDhs', $bigtypeDhs);                          //将查询结果赋给模板变量
if($requestPageName == 'index.php'){                                 //根据当前文件名称为变量$db 赋值
     $dh = 'index';
}elseif ($requestPageName == 'listsepbook.php'){
     if($_GET['t'] == 'new'){
          $dh = 'new';
     }elseif ($_GET['t'] == 'sepprice'){
          $dh = 'sepprice';
     }elseif ($_GET['t'] == 'hotsell'){
          $dh = 'hotsell';
     }elseif ($_GET['t'] == 'term'){
          $dh = 'term';
     }
}elseif ($requestPageName == 'listreadbook.php'){
     $dh = 'listread';
}
$smarty->assign('dh', $dh);                                          //将变量值传递到模板变量
if($requestPageName == 'search.php'){
     if($_GET['stype'] == 'simple'){
          $keytxt = urldecode(str_replace('@', '%', $_GET['keyWord']));
     }elseif ($_POST['stype'] == 'simple'){
          $keytxt = $_POST['keyWord'];
     }
     $smarty->assign('keytxt', $keytxt);
}
$smarty->display ( 'header.phtml' );                                 //指定模板页
```

（2）创建模板页 views\header.phtml，设计网站头文件的页面，输出模板变量中传递的数据，其代码请参考本书光盘，这里不再赘述。

秘笈心法

心法领悟 591：通过 include 语句包含网站的头文件。



```
$systemBottom = $adminDB->executeSQL("select bq, icp from tb_system where mark = 1", $connID);
$smarty->assign('systemBottom', $systemBottom);
$smarty->display('footer.phtml');
$connDB->closeConnID();
```

（2）在 views 文件夹下新建 footer.phtml 模板文件，输出模板变量中存储的系统信息，同时创建常用功能的超链接，其代码请参考本书光盘。

秘笈心法

心法领悟 592：应用 require_once 语句包含文件。

require_once 语句是 require 语句的延伸，其功能与 require 语句基本类似，不同的是，在应用 require_once 语句时会先检查要包含的文件是不是已经在该程序中的其他地方被包含，如果是，则不会再次重复调用该文件。

include_once 语句和 require_once 语句的用途是确保一个被包含文件只能被包含一次。使用这两个语句可以防止意外地多次包含相同的函数库，避免因函数的重复定义产生的错误。但两者之间是有区别的：include_once 语句在脚本执行期间调用外部文件发生错误时产生一个警告，而 require_once 语句则导致一个致命错误。

include_once 和 require_once 语句在大小写不敏感的操作系统中（如 Windows）的行为可能不是所期望的。

实例 593	首页广告设计 光盘位置：光盘\MR\15\593	中级 趣味指数：★★★

实例说明

明日科技图书网的首页中有一个循环播放的广告位，本实例实现 3 个广告的循环播放，并且每个广告链接不同的文件，其运行效果如图 15.18 所示。



图 15.18　首页广告设计

关键技术

首页广告的设计应用的是 js 脚本技术，控制图片的展开方法，完成图片的循环播放，其调用的是一个已经定义好的 js 脚本 lzhScrollImg.js。

在模板页中还要设置一些参数，关键代码如下：

```
<script language="javascript">
        var w = "515px";                                          //广告宽度
        var h = "150px";                                          //广告高度
        var bg = "#99CCFF";                                       //广告背景色
        var imgSrc = new Array;                                   //图片地址
        imgSrc[0] = '{$system[0].ggurl}/banner_0.jpg';
        imgSrc[1] = '{$system[0].ggurl}/banner_1.jpg';
        imgSrc[2] = '{$system[0].ggurl}/banner_2.jpg';
        var imgLink = new Array;                                  //图片链接地址
        imgLink[0] = 'http://www.mrbooks.cn';
        imgLink[1] = 'http://www.mrbccd.com';
        imgLink[2] = 'http://www.mrbooks.cn/listsepbook-seppрrice.html';
        var imgLinkTitle = new Array;                             //图片链接提示文字
        imgLinkTitle[0] = '明日电脑书城';
        imgLinkTitle[1] = '编程词典';
        imgLinkTitle[2] = '优惠购书';
</script>
<script language="javascript" src="{util->baseUrl}/js/lzhScrollImg.js"></script>
```

设计过程

（1）在 index.phtml 模板页中，定义 JavaScript 脚本，定义广告位的各种参数并且载入调用的 js 脚本。其代码请参考关键技术中的内容。



```
                if (isset($_SESSION["unc"]) && $_SESSION["unc"] != "") {
                    unset($_SESSION["unc"]);
                }                                               //销毁原来登录用户的登录昵称
                session_register("unc");                        //注册新的用户昵称到 SESSION 变量
                $_SESSION["unc"] = trim($_POST["usernc"]);
            if (isset($_SESSION['toUrl'])) {                    //获取要跳转的页面，并进行跳转
                    echo "<script>window.location.href='" . $util->baseUrl() . "/" . $_SESSION['toUrl'] . "';</script>";
                } else {
                    echo "<script>window.location.href='" . $util->baseUrl() . "/regsuccess.html';</script>";
                }
            } else {
                echo "<script>alert('注册失败，请重试！')</script>";
            }
        } else {
            echo "<script>alert('该昵称已经被他人注册！')</script>";
        }
}
$smarty->display('getuserinfo.phtml');
require_once 'footer.php';
```

秘笈心法

心法领悟 594：防止用户越过表单限制上传数据。

在本实例的 getuserinfo.php 文件中，为了避免由于用户的浏览器禁用 JavaScript 脚本功能而使用户输入昵称的唯一性验证失效，或者某些用户越过表单的验证机制直接上传数据，在保存用户提交的注册信息前，首先使用数据库管理类的 executeSQL()方法查询是否存在和用户提交昵称相同的注册用户，如果存在则提示该昵称已被占用，并返回注册页面要求用户重新填写注册信息，反之则将用户提交的注册信息保存到数据库中。

实例 595	用户登录 光盘位置：光盘\MR\15\595	高级 趣味指数：★★★☆

实例说明

用户在明日科技图书网购买图书时，需要首先登录站点，这样可以记录购买人信息，并且可在用户中心的订单查询模块中查询到该用户的所有订单信息。明日科技图书网的用户登录页面如图 15.20 所示。

图 15.20　用户登录页面

关键技术

登录的目的就是为用户开据一张通行证，使其可以在网站中进行任意操作（权限范围之内），也是其行使网站会员权限的依据。而这个通行证的开据应用的是 SESSION 技术。登录成功后，将用户名存储到 SESSION 变量中，即完成通行证的开据，其代码如下：

```
$_SESSION['unc'] = $usernc;
```

注意： 在应用 SESSION 变量之前，切记一定要应用 session_start()函数初始化 SESSION 变量，否则 SESSION 变量不能使用，并且在应用 session_start()函数之前不能有任何输出。本实例中 SESSION 变量的初始化操作是在包含文件 lzh.inc.php 中完成的。

该功能利用 SESSION 可以在页面之间传递数据的特性，确保这个通行证在网站的任意一个页面中都有效，直到用户退出登录。

设计过程

（1）在 views 文件夹下创建 login.phtml 登录表单的 UI 界面，其实现方法与注册表单类似，这里不再赘述。

（2）当用户单击“登录”按钮后，将对用户输入的用户名、密码和验证码进行有效性验证，如果验证通过则将用户输入的数据提交到 login.php 中进行处理，关键代码如下：

```
<?php
require_once 'header.php';
$errorMsg = '';
if (isset($_POST['usernc']) && $_POST['usernc'] != '') {                //判断是否提交了表单
    $usernc = trim($_POST['usernc']);                                   //获得提交的用户名
    $pwd = trim($_POST['userpwd']);                                     //获得提交的登录密码
    $user = $adminDB->executeSQL("select id, usernc from tb_user where usernc='" . $usernc . "' and pwd='" . md5($pwd) . "'", $connID); //按用户名和密码查询用户
    if (! $user) {
        $errorMsg = '用户名或密码输入有误！ ';
    } else {
        if (isset($_SESSION["unc"])) {                                  //销毁原登录用户信息，并注册新登录用户信息给 SESSION 变量
            session_unregister('unc');
        }
        $_SESSION['unc'] = $usernc;
        @$adminDB->executeSQL("update tb_user set logintimes=logintimes+1 ,lastlogintime ='" . date('Y-m-d H:i:s') . "',ip='" .
$_SERVER['REMOTE_ADDR'] . "' where usernc='" . $_SESSION['unc'] . "'", $connID);          //更改用户的登录信息
        if (isset($_SESSION['toUrl'])) {
            echo "<script>window.location.href='" . $util->baseUrl() . "/" . $_SESSION['toUrl'] . "';</script>";
            exit();
        } else {
            echo "<script>window.location.href='" . $util->baseUrl() . "/usercenter.html';</script>";
            exit();
        }
    }
}
$smarty->assign('errorMsg', $errorMsg);
$smarty->display('login.phtml');
require_once 'footer.php';
```

上述代码中，首先定义变量$errorMsg，保存处理登录表单提交数据时出现的错误信息（例如登录失败信息等），然后使用$_POST 全局数组接收用户提交的用户名和登录密码，并使用数据库管理类的 executeSQL()方法执行查询语句来判断用户输入的用户名和密码是否正确，如果正确则更改用户登录次数和最后登录时间，反之则将登录失败信息赋给变量$errorMsg，并传递给 login.phtml 模板。

秘笈心法

心法领悟 595：延长 SESSION 的生命周期。

在开发程序的过程中可能会遇到这样的情况，以会员的身份登录一个网站后，如停留的时间较长，此时在这个网站中发布信息时，会提示“您没有登录，请登录后再发布信息”。

之所以出现此类问题，就是因为在登录时定义的 SESSION 变量已经过期，导致当前用户不可用，需要重新登录。解决这个问题最好的方法就是通过 session_set_cookie_params()函数延长 SESSION 的生命周期，将过期时间设置为 0。

实例 596	用户中心 光盘位置：光盘\MR\15\596	高级 趣味指数：★★★★☆

实例说明

明日科技图书网的用户中心，主要用于实现对用户注册信息和用户登录密码的修改、用户订单的查询和意

本框、新密码录入文本框和确认密码录入文本框 3 个表单元素，如图 15.22 所示。

系统将用户修改的数据提交到 usercenterchangepwd.php 页面，在该页面中完成对密码的更改。首先判断用户是否非法登录，然后判断用户输入的原密码是否正确，如果正确则将用户密码更改成新值，反之给出错误提示，其代码如下：

```
<?php
require_once 'header.php';
if (! isset($_SESSION['unc']) || $_SESSION['unc'] == '') {                    //判断用户是否已经登录
    echo "<script>alert('禁止非登录！');window.location.href='" . $util->baseUrl() . "/index.html';</script>";
    exit();
}
if (isset($_POST['oldpwd']) && $_POST['oldpwd'] != '') {                      //判断用户是否已经提交了表单
    if (! $adminDB->executeSQL("select id, usernc from tb_user where usernc='" . $_SESSION['unc'] . "' and pwd='" . md5($_POST['oldpwd']) . "'",
$connID)) {
        echo "<script>alert('原密码输入有误！');</script>";                  //验证原密码是否正确
    } else {
        if ($adminDB->executeSQL("update tb_user set pwd='" . md5($_POST['pwd1']) . "', truepwd='" . $_POST['pwd1'] . "' where usernc='" .
$_SESSION['unc'] . "'", $connID)) {                                           //执行更改密码操作
            echo "<script>alert('密码更改成功！');</script>";
        } else {
            echo "<script>alert('密码更改失败！');</script>";
        }
    }
}
$smarty->display('usercenterchangepwd.phtml');
require_once 'footer.php';
```

（3）意见反馈。

用户在参与本站活动时，如果对网站的管理或运营存在建议或意见，可以通过用户中心的意见反馈模块将个人意见提交给网站管理者。明日科技图书网的意见反馈表单如图 15.23 所示。

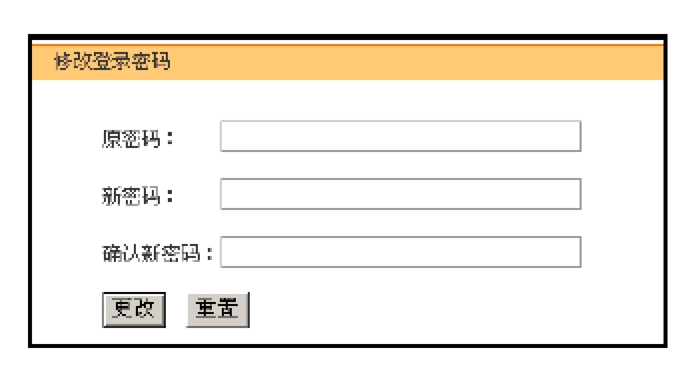


图 15.22　用户密码修改表单

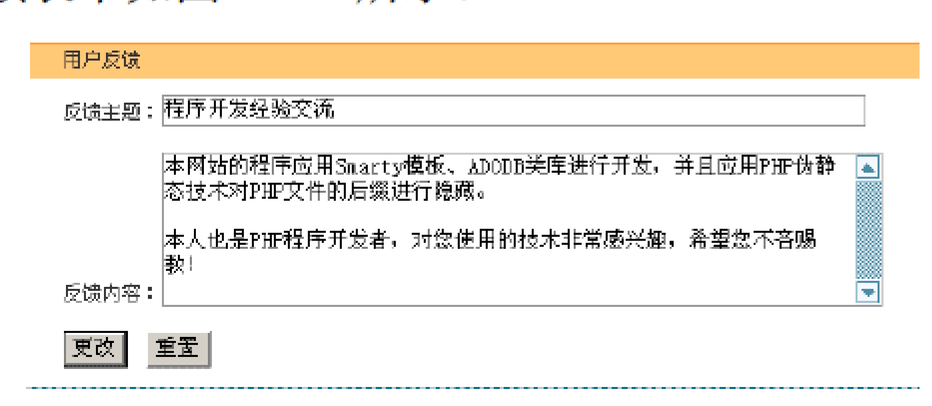


图 15.23　用户意见反馈表单

用户填写的反馈意见将被提交到 userfeedback.php 页面，在该页面中应用数据库管理类中的 executeSQL()方法将反馈意见添加到 tb_feedback 数据表中，其代码如下：

```
if (! isset($_SESSION['unc']) || $_SESSION['unc'] == '') {                    //判断用户是否已经登录
    echo "<script>alert('禁止非登录！');window.location.href='" . $util->baseUrl() . "/index.html';</script>";
    exit();
}
if (isset($_POST['title']) && $_POST['title'] != '') {                        //判断用户是否已经提交了表单
    if ($adminDB->executeSQL("insert into tb_feedback(title, content, addtime, usernc) values('" . $_POST['title'] . "', '" . $_POST['content'] . "', '" .
date('Y-m-d H:i:s') . "', '" . $_SESSION['unc'] . "')", $connID)) {           //保存用户的反馈意见信息
        echo "<script>alert('您的反馈信息已经成功保存，谢谢您的支持！');</script>";
    } else {
        echo "<script>alert('反馈信息保存失败，请重试！');</script>";
    }
}
```

秘笈心法

心法领悟 596：用户中心功能总结。

在用户中心功能模块中应用的技术可以划分为两部分：第一部分，在 Smarty 模板页中创建 form 表单，提交用户需要更新的数据；第二部分，在 PHP 动态页中，应用数据库管理类中的 executeSQL()方法执行 select、insert 和 update 语句，完成对用户的验证、数据的添加和更新操作。

```
//如果变量 orderno 的值不为空，并且变量 def 的值不等于 qx，则执行查询语句，计算出订单的详细信息，并且将查询结果赋给模板变量
if (isset($_GET['orderno']) && $_GET['orderno']!='' && $_GET['def'] !='qx'){
     $order = $adminDB->executeSQL("select id, orderno, username, address, sex, yb, tel, idstr, numstr, rectype, paytype, goodsprice, yjprice, totalprice,
addtime, isfk, isfh, issh from tb_order where orderno='".trim($_GET['orderno'])."'", $connID);
          $arrayIds = explode('@', $order[0]['idstr']);
          $arrayNums = explode('@', $order[0]['numstr']);
          $arrayCarInfos = array();
          $totalPrice = 0;
          for($i = 0; $i < count($arrayIds); $i++){
              $bookid = $arrayIds[$i];
              if($bookid != ''){
                   $tmpArray = array();
                   $bookinfo = $adminDB->executeSQL("select id, bookname ,oldprice, newprice from tb_bookinfo where id='".$bookid."'", $connID);
                   $tmpArray['id'] = $bookinfo[0]['id'];
                   $tmpArray['bookname'] = $bookinfo[0]['bookname'];
                   $tmpArray['oldprice'] = $bookinfo[0]['oldprice'];
                   $tmpArray['newprice'] = $bookinfo[0]['newprice'];
                   $tmpArray['num'] = $arrayNums[$i];
                   $tmpArray['smallTotalPrice'] = $bookinfo[0]['newprice'] * $arrayNums[$i];
                   $totalPrice += $tmpArray['smallTotalPrice'];
                   array_push($arrayCarInfos, $tmpArray);
              }
         }
          $smarty->assign('arrayCarInfos', $arrayCarInfos);
          $smarty->assign('order', $order);
          $isShow = 'T';
}
$smarty->assign('isShow', $isShow);
//执行查询操作，输出当前用户的订单信息
$orders = $adminDB->executeSQL("select orderno, addtime, username, totalprice from tb_order where orderusernc='".$_SESSION['unc']."' and isqx=0",
$connID);
$smarty->assign('orders', $orders);                    //将查询结果赋给模板变量
$smarty->display('usercenterorder.phtml');             //指定模板页
require_once 'footer.php';                             //包含网站的尾文件
```

（2）在 views 文件夹下创建 usercenterorder.phtml 模板文件，根据模板变量中传递的数据完成“我的订单”模块页面的设计。首先，通过 section 语句循环输出当前用户的订单信息并创建“查看”超链接链接到本页，将订单号作为参数进行传递。然后，应用 if 语句判断当模板变量$isShow 的值为 T 时，输出模板变量中传递的订单详细信息。最后，创建“取消订购”和“确认收货”按钮，通过 onclick 事件调用 JavaScript 完成对订单的处理操作，其关键代码如下：

```
{section name=oID loop=$orders}
<div style="width:90%; height:20px; border-bottom:1px solid #0C92AB; border-left:1px solid #0C92AB; border-right:1px solid #0C92AB;
padding-top:3px">
<li style="display:inline; width:26%; float:left; border-right:1px solid #0C92AB">{$orders[oID].orderno}</li>
     <li style="display:inline; width:22%; float:left; border-right:1px solid #0C92AB">{$orders[oID].username}</li>
     <li style="display:inline; width:22%; float:left; border-right:1px solid #0C92AB">{util->moneyFormat p=$orders[oID].totalprice}</li>
     <li style="display:inline; width:22%; float:left; border-right:1px solid #0C92AB">{$orders[oID].addtime}</li>
     <li style="display:inline; width:8%; float:left"><a href="{util->baseUrl}/usercenterorder-{$orders[oID].orderno}.html" class="a1">查看</a></li>
</div>
{/section}
{if $isShow == 'T'}
<div style="width:680px; border-top:1px dotted #CCCCCC; border-bottom:1px dotted #CCCCCC; padding:15px">
        <div style="width:630px; height:22px; text-align:left; padding-top:5px">
                <li style="display:inline; width:200px; height:22px; background-color:#006D84; padding-top:5px; padding-left:10px; color:#FFFFFF;
float:left"> <strong>订单号：{$order[0].orderno}</strong></li>
                <li style="display:inline; width:200px; height:22px; padding-top:5px; float:right">订购时间：{$order[0].addtime}</li>
        </div>
<!--省略了部分代码-->
        <div style="width:680px; height:30px; padding-top:3px; text-align:left"><strong> 订购状态</strong>   [ 已收款：
<input type="checkbox" disabled {if $order[0].isfk == 1 }checked{/if} />   已发货：
<input type="checkbox" disabled {if $order[0].isfh == 1 }checked{/if}/> 已收货：
<input type="checkbox" disabled {if $order[0].issh == 1 }checked{/if}/> ]
<input type="button" value="取消订购" {if $order[0].issh == 1 || $order[0].isfk == 1 || $order[0].isfh == 1 }disabled{/if}
onclick="javascript:if(window.confirm('您确认取消该订单吗？')==true){ldelim}window.location.href='usercenterorder-{$order[0].orderno}
-qx.html';{rdelim}"/>
                <input type="button" value="确认收货" onclick="javascript:if(window.confirm('您确认已经收货了吗？')==true){ldelim}
window.location.href='usercenterorder-{$order[0].orderno}-y.html';{rdelim}" {if $order[0].issh == 1 }disabled{/if}/>
        </div>
```

```
</div>
{/if}
```

秘笈心法

心法领悟 597：实现退出登录功能。

用户在登录成功后，可以修改注册信息、登录密码，查看订单信息和提交意见反馈等，那么如何实现退出登录呢？

实现退出登录功能应用的是 session_start()、session_unregister()、unset()和 header()函数。首先初始化 SESSION 变量，然后注销 SESSION 变量 unc，接着删除 SESSION 变量，最后跳转到 index.html 页面，该功能存储于 logout.php 文件中。

实例 598	找回密码 光盘位置：光盘\MR\15\598	高级 趣味指数：★★★☆

实例说明

在明日科技图书网的用户登录页面中，有一个“找回密码”超链接，单击该链接将执行密码找回的操作。第一步：输入用户的昵称；第二步：输入密码提示答案；第三步：如果输入答案不正确，则给出错误提示信息，否则弹出密码找回成功对话框并且显示原密码，也可以选择重新设置密码，其运行效果如图 15.25 所示。

图 15.25　找回密码的操作流程

关键技术

在找回密码功能实现过程中，应用到的技术如下：

（1）应用 Smarty 模板中的 ldelim 和 rdelim 标签输出 Smarty 模板中的定界符，实现直接在 Smarty 模板页中定义 JavaScript。

（2）应用 window 对象中的 open()方法打开 openfindpwd.php 文件，开始执行找回密码的操作。

（3）应用 PHP 中提供的 MySQL 数据库函数操作 MySQL 数据库，完成找回密码、更改密码的操作。其中，首先在 conn.php 文件中应用 mysql_connect()、mysql_select_db()和 mysql_query()函数完成与数据库服务器、数据库的连接，并且设置编码格式，其代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_database15",$conn);                 //连接数据库
mysql_query("set names gb2312");                        //设置数据库编码格式
?>
```

设计过程

（1）在模板页 login.phtml 中应用 Smarty 模板中的 ldelim 和 rdelim 标签直接编写 JavaScript 脚本方法 openfindpwd()，打开 openfindpwd.php 文件，其关键代码如下：

```
<script language="JavaScript" type="text/javascript">
function openfindpwd()
{ldelim}
window.open("openfindpwd.php","newframe","left=200,top=200,width=200,height=100,menubar=no,toolbar=no,location=no,scrollbars=no,location=no
");{rdelim}
</script>
<a href="javascript:openfindpwd()" class="a1">[找会密码]</a>
```

（2）新建 openfindpwd.php 文件，创建 form 表单，提交要找回密码的用户的昵称，并且通过 JavaScript 脚



本判断用户提交的数据是否为空，将数据提交到 findpwd.php 页面。

（3）新建 findpwd.php 文件。首先包含数据库连接文件 conn.php，然后判断用户提交的昵称在数据库中是否存在，如果存在则输出此用户注册时填写的密码提示问题，并且通过 form 表单将问题的答案提交到 showpwd.php 文件中进行处理。

（4）新建 showpwd.php 文件，首先包含数据库的连接文件 conn.php，然后判断用户提交的问题答案是否正确，如果不正确则给出提示信息并跳转到上一页；如果正确则输出显示原密码和重设密码的超链接。

（5）新建 changepwd.php 文件，创建重新设置密码的 form 表单，将重新设置的密码提交到 savechangepwd.php 文件中。

（6）新建 savechangepwd.php 文件，完成用户密码的更新，其代码如下：

```
<?php
include_once ("conn.php");                      //连接数据库
$truepwd = $_POST[userpwd1];                    //获取提交的新密码
$pwd = md5($truepwd);                           //对密码进行 md5 加密
if (mysql_query("update tb_user set pwd='$pwd',truepwd='$truepwd' where id='" . $_POST["userid"] . "'", $conn)) {   //执行更新操作
    echo "<script>alert('密码更改成功!');history.back();</script>";
} else {
    echo "<script>alert('密码更改失败!');history.back();</script>";
}
mysql_close($conn);                             //关闭数据库连接
?>
```

秘笈心法

心法领悟 598：MySQL 数据库函数。

在明日科技图书网站项目中，为了体现本章内容的综合性，不但应用 Smarty 模板和 ADODB 类库技术，同时也融入了 MySQL 函数操作 MySQL 数据库的方法。有关通过 MySQL 函数操作 MySQL 数据库的详细讲解可以参考第 8 章的内容。

15.4　数据的输出

本节将介绍数据输出功能的实现，包括图书导航、图书分类、特别图书、图书试读、图书详细信息和新闻公告等功能。

实例 599	图书导航 光盘位置：光盘\MR\15\599	高级 趣味指数：★★★☆

实例说明

明日科技图书网站中，通过图书导航功能展示出网站中图书的所有类别，并且将图书划分为初级、中级和高级 3 种类型，图书导航功能的运行效果如图 15.26 所示。



图 15.26　图书导航功能

关键技术

实现图书导航功能应用的关键技术如下：

（1）在 index.php 动态页中，应用数据库管理类中的 executeSQL()方法查询出图书大类和小类数据表中的数据。其中关于数据库的连接和操作方法可以参考本章实例 586 和实例 587。

（2）在 index.phtml 模板页中，应用 Smarty 模板中的 if 语句根据模板变量传递的值判断应具体输出哪种类型的数据，然后应用 section 语句进行嵌套，完成图书大类与小类的对应循环输出。

以分页的形式展现出来。明日科技图书网图书分类展示效果如图 15.27 所示。



图 15.27　明日科技图书网图书分类展示效果

关键技术

实现图书分类展示的关键是数据表的创建，在图书大类表 tb_bigtype 和图书小类表 tb_smalltype 之间存在一个关联的字段，通过该字段确立两个表中数据的从属关系。这里将 tb_bigtype 表中的 id 字段与 tb_smalltype 表中的 bigtypeid 字段建立关联，即在 tb_smalltype 表中，根据字段 bigtypeid 的值可以确定该小类属于哪个大类。

设计过程

在图书分类展示页面中，浏览者可以从最上面的图书大类中选择要购买图书的大类，然后在子类中选择具体的语言类别，这样就可以将指定语言类别的图书展示出来，同时可以按照指定的排列方式和图书层次对图书的显示顺序进行调整。具体实现过程如下：

（1）在 listbook.php 文件中，首先使用$_GET 全局数组接收传递到页面的排序方式标识、图书层次标识和当前页显示的页码，然后使用分页类的 pageData()方法获取分页参数，最后指定模板页，其关键代码如下：

```
if (! isset($_GET['pltype']) || $_GET['pltype'] == '') {                    //获得图书排列方式
    $pltype = '1';
} else {
    $pltype = $_GET['pltype'];
}
$smarty->assign('pltype', $pltype);
if (! isset($_GET['cctype']) || $_GET['cctype'] == '') {                    //获得图书层次标识
    $cctype = '1';
} else {
    $cctype = $_GET['cctype'];
}
$smarty->assign('cctype', $cctype);
if (! isset($_GET['page']) || $_GET['page'] == '') {                        //获得当前要显示的页码
    $page = '1';
} else {
    $page = $_GET['page'];
}
$smarty->assign('page', $page);
$smalltype = $adminDB->executeSQL("select id, bigtypeid, typename from tb_smalltype where id='$stid'", $connID);          //查询图书小类
$smarty->assign('smalltype', $smalltype);
$bigtype = $adminDB->executeSQL("select id, typename from tb_bigtype where id='" . $smalltype[0]['bigtypeid'] . "'", $connID);
$smarty->assign('bigtype', $bigtype);
$smalltypes = $adminDB->executeSQL("select id, typename from tb_smalltype where bigtypeid='" . $bigtype[0]['id'] . "'", $connID);   //查询图书大类
$smarty->assign('smalltypes', $smalltypes);
if ($cctype == '1') {                                                       //设置图书层次条件
    $where = " and smalltypeid = '" . $stid . "'";
} elseif ($cctype == '2') {
    $where = " and bookcc = 1 and smalltypeid = '" . $stid . "'";
```

```
} elseif ($cctype == '3') {
    $where = " and bookcc = 2   and smalltypeid = '" . $stid . "'";
} elseif ($cctype == '4') {
    $where = " and bookcc = 3   and smalltypeid = '" . $stid . "'";
}
if ($pltype == '1') {                                                          //设置图书排列方式
    $order = "addtime desc";
} elseif ($pltype == '2') {
    $order = "pubtime desc";
} elseif ($pltype == '3') {
    $order = "newprice asc";
} elseif ($pltype == '4') {
    $order = "browsertime desc";
}
$sql = "select tb_bookinfo.id, tb_bookinfo.bookimg, tb_bookinfo.oldprice, tb_bookinfo.newprice, tb_bookinfo.about, tb_bookinfo.browsertime,
tb_bookinfo.bookname, tb_bookinfo.writer, tb_bookinfo.pubtime, tb_bookinfo.addtime, tb_bookinfo.ishave, tb_pub.pubname from tb_bookinfo, tb_pub
where tb_bookinfo.pubid = tb_pub.id " . $where . " order by " . $order . "";          //构建查询语句
$bookinfos = $pageDB->pageData($sql, $connID, 10, $page);                    //执行分页查询
$smarty->assign('bookinfos', $bookinfos);
```

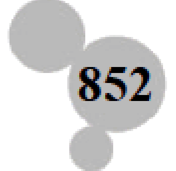
（2）在 views\listbook.phtml 模板页中，应用 section 语句完成指定类别图书信息的分页显示，关键代码如下：

```
{if $bookinfos.countRs > 0}
{section name=pbID loop=$bookinfos.data}
<div style="width:930px">
    <br>
    <div style="width:100%; height:180px">
        <div style="width:20%; height:100%; float:left">
            <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{$system[0].bookimgurl}/
{$bookinfos.data[pbID].bookimg}" width="110" height="150" style="border:1px solid #123456" /></a>
        </div>
        <div style="width:78%; height:100%; float:right; text-align:left; font-size:13px">
            <li style="display:inline"><a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html" class="a5"><strong>
{$bookinfos.data[pbID].bookname}</strong></a></li>
            <br><br>
            <li style="display:inline">出版社：{$bookinfos.data[pbID].pubname}</li>
            <br><br>
            <li style="display:inline">作者：{$bookinfos.data[pbID].writer}</li>
            <br><br>
            <li style="display:inline">上架时间：{$bookinfos.data[pbID].addtime|truncate:10:""}  出版时间：
{$bookinfos.data[pbID].pubtime|truncate:10:""}   浏览次数：{$bookinfos.data[pbID].browsertime} 次</li>
            <br><br>
            <li style="display:inline">市场价：<img src="{util->baseUrl}/img/mark_green.gif" /> <s>{util->moneyFormat
p=$bookinfos.data[pbID].oldprice}</s>  <font color="#FF0000">会员价：<img src="{util->baseUrl}/img/mark_red.gif"
/> {util->moneyFormat p=$bookinfos.data[pbID].newprice}</font>
             </li>
             <br><br>
             <li style="display:inline"><a href="{util->baseUrl}/cart-{$bookinfos.data[pbID].id}-add.html"><img src="{util->baseUrl}/
img/button_order.gif" border="0"/></a> <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{util->baseUrl}/img/
button_info.gif" border="0"/></a></li>
        </div>
      </div>
    <div style="width:95%; height:50px; border:1px solid #CCCCCC; text-align:left; line-height:18px; color:#990000; padding:5px">
     {util->unHtml p=$bookinfos.data[pbID].about|truncate:420:"...":false}
    </div>
</div>
{/section}
{else}
<div style="width:500px; height:80px; font-size:14px; color:#FF0000">
    <img src="{util->baseUrl}/img/mark_telllogin.gif"><br />
            <strong>对不起，暂无该类图书！</strong>
</div>
{/if}
```

秘笈心法

心法领悟 600：为什么编辑的模板页文件后缀是.phtml，而输出时文件的后缀是.html。

因为本实例的开发应用 PHP 伪静态技术，会对输出文件的后缀和传递的参数进行重新定义。伪静态技术的执行要通过.htaccess 文件。



设计过程

（1）建立 listsepbook.php 文件用来实现业务逻辑。在该文件中，首先使用$_GET 全局数组接收查询字符串参数 t 的值，然后根据该值决定当前页面要显示的内容并构建查询语句，最后使用分页类的 pageData()方法实现分页查询，其完整代码请参考本书光盘，这里不再赘述。

（2）建立模板页 listsepbook.phtml，通过 section 语句分页输出指定类别下的图书信息，其关键代码如下：

```
{if $bookinfos.countRs > 0}
{section name=pbID loop=$bookinfos.data}
<div style="width:930px">
      <br>
      <div style="width:100%; height:180px">
            <div style="width:20%; height:100%; float:left">
                  <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{$system[0].bookimgurl}/
{$bookinfos.data[pbID].bookimg}" width="110" height="150" style="border:1px solid #123456" /></a>
            </div>
            <div style="width:78%; height:100%; float:right; text-align:left; font-size:13px">
                  <li style="display:inline"><a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html" class="a5"><strong>
{$bookinfos.data[pbID].bookname}</strong></a></li>
                  <br><br>
                  <li style="display:inline">出版社：{$bookinfos.data[pbID].pubname}</li>
                  <br><br>
                  <li style="display:inline">作者：{$bookinfos.data[pbID].writer}</li>
                  <br><br>
                  <li style="display:inline">上架时间：{$bookinfos.data[pbID].addtime|truncate:10:""}      出版时
间：{$bookinfos.data[pbID].pubtime|truncate:10:""}      浏览次数：{$bookinfos.data[pbID].browsertime} 
次</li>
                  <br><br>
                  <li style="display:inline">市场价：<img src="{util->baseUrl}/img/mark_green.gif" /> <s>{util->moneyFormat
p=$bookinfos.data[pbID].oldprice}</s>            
                        <font color="#FF0000">会员价：<img src="{util->baseUrl}/img/mark_red.gif" /> {util->moneyFormat
p=$bookinfos.data[pbID].newprice}</font>
                   </li>
                   <br><br>
                   <li style="display:inline"><a href="{util->baseUrl}/cart-{$bookinfos.data[pbID].id}-add.html"><img
src="{util->baseUrl}/img/button_order.gif" border="0"/></a>    <a
href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{util->baseUrl}/img/button_info.gif" border="0"/></a></li>
            </div>
                  </div>
      <div style="width:95%; height:50px; border:1px solid #CCCCCC; text-align:left; line-height:18px; color:#990000; padding:5px">
      {util->unHtml p=$bookinfos.data[pbID].about|truncate:420:"...":false}
      </div>
</div>
{/section}
{else}
<div style="width:500px; height:80px; font-size:14px; color:#FF0000">
      <img src="{util->baseUrl}/img/mark_tellogin.gif"><br />
                  <strong>对不起，暂无新书信息！</strong>
</div>
{/if}
```

秘笈心法

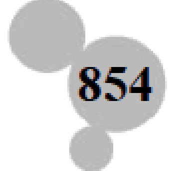
心法领悟 601：SQL 语句中的多表查询。

在 SQL 语言中，通过 where 子句实现多表查询，所要查找的字段名用“表明.字段名”表示，这样可以避免因表之间字段重名而无法获知该字段属于哪个表。在 where 子句中多个表之间所形成的联动关系应按如下格式书写：

```
表 1.字段=表 2.字段  and  其他查询条件
```

多表查询的语法如下：

```
select  字段名  from  表 1,表 2... where  表 1.字段=表 2.字段  and  其他查询条件
```



实例 602　**图书试读**　光盘位置：光盘\MR\15\602　高级　趣味指数：★★★☆

实例说明

为了能够让读者了解某本图书的大概内容，吸引更多读者购买图书，在制作明日科技图书网时，将图书试读作为单独模块，其运行效果如图 15.29 所示。



图 15.29　试读图书展示

关键技术

实现图书试读的关键技术是如何获取到试读文件，本实例中应用 HTTP 方式完成试读文件的下载操作，其中关键是 header()函数的应用。有关应用 HTTP 方式进行下载的详细讲解请参考本书第 4 章的内容。

在本项目中，下载的操作在 upfiles\doc\download.php 文件中完成，同时还对文件被下载的次数进行统计，其关键代码如下：

```
<?php
require_once 'lzh.inc.php';                    //包含类的实例化文件
$filename=$_GET['filename'];                   //获取文件名称
$rid = $_GET['rid'];                           //获取文件的 id
$system = $adminDB->executeSQL("update tb_read set downtimes = downtimes+1 where id='$rid'", $connID);     //执行更新操作
if(file_exists($filename)==false){             //判断上传文件是否存在
  echo "<script>alert('对不起，本站暂时停止该文件下载!');history.back();</script>";
  exit;
}
$fp=fopen($filename,"r");                      //打开上传文件
header("Content-type:application/octet-stream");
header("Accept-ranges:bytes");
header("Accept-length:".filesize($filename));
header("Content-Disposition:attachment;filename=".$filename);
echo fread($fp,filesize($filename));           //读取上传文件内容
fclose($fp);                                   //关闭文件
```

设计过程

图书试读模块对明日科技图书网站中所有可以试读的图书进行分页展示。

（1）新建 listreadbook.php 文件，完成所有试读图书的分页查询操作。首先通过$_GET 全局数组接收查询字符串 page 的值，该值用来指定当前页面要显示的内容。然后构建查询语句，通过分页类的 pageData()方法获得分页数据。最后将查询结果赋给模板变量并且指定模板页，具体实现代码如下：

```
require_once 'header.php';
$nowtype = '图书试读下载';
$smarty->assign('nowtype', $nowtype);
$system = $adminDB->executeSQL("select bookimgurl, readurl from tb_system where mark = 1", $connID);
$smarty->assign('system', $system);                              //查询系统配置信息
if (! isset($_GET['page']) || $_GET['page'] == '') {              //获取当前页码
    $page = '1';
} else {
    $page = $_GET['page'];
}
$sql = "select tb_bookinfo.id, tb_bookinfo.bookimg, tb_bookinfo.oldprice, tb_bookinfo.newprice, tb_bookinfo.about, tb_bookinfo.browsertime,
tb_bookinfo.bookname, tb_bookinfo.writer, tb_bookinfo.pubtime, tb_bookinfo.addtime, tb_bookinfo.ishave, tb_pub.pubname, tb_read.id as readid,
tb_read.filename from tb_bookinfo, tb_pub, tb_read where tb_bookinfo.pubid = tb_pub.id and tb_read.bookinfoid = tb_bookinfo.id order by
tb_read.addtime desc";                                            //构建查询语句
$bookinfos = $pageDB->pageData($sql, $connID, 10, $page);         //执行分页查询
$smarty->assign('bookinfos', $bookinfos);
$smarty->display('listreadbook.phtml');
require_once 'footer.php';
```

（2）新建模板文件 listreadbook.phtml，应用 section 语句完成试读图书的分页输出，创建“试读下载”超链接，链接到 upfiles\doc\download.php 文件，完成试读文件的下载操作，其关键代码如下：

```
{if $bookinfos.countRs > 0}
{section name=pbID loop=$bookinfos.data}
<div style="width:930px">
    <br>
    <div style="width:100%; height:180px">
        <div style="width:20%; height:100%; float:left">
            <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{$system[0].bookimgurl}/
{$bookinfos.data[pbID].bookimg}" width="110" height="150" style="border:1px solid #123456" /></a>
        </div>
          <div style="width:78%; height:100%; float:right; text-align:left; font-size:13px">
            <li style="display:inline"><a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html" class="a5"><strong>
{$bookinfos.data[pbID].bookname}</strong></a></li>
            <li style="display:inline"><a href="{util->baseUrl}/cart-{$bookinfos.data[pbID].id}-add.html"><img src="{util->baseUrl}/
img/button_order.gif" border="0"/></a>    <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img
src="{util->baseUrl}/img/button_info.gif" border="0"/></a>    <a href="{$system[0].readurl}/download.php?rid=
{$bookinfos.data[pbID].readid}&filename={$bookinfos.data[pbID].filename}"><img src="{util->baseUrl}/img/button_sdcz.gif" border="0"/></a></li>
        </div>
       </div>
    <div style="width:95%; height:50px; border:1px solid #CCCCCC; text-align:left; line-height:18px; color:#990000; padding:5px">
    {util->unHtml p=$bookinfos.data[pbID].about|truncate:420:"...":false}
    </div>
</div>
{/section}
{else}
<div style="width:500px; height:80px; font-size:14px; color:#FF0000">
    <img src="{util->baseUrl}/img/mark_telllogin.gif"><br />
            <strong>对不起，暂无试读图书信息！</strong>
</div>
{/if}
```

（3）有关下载试读文件的内容请参考本实例的关键代码，这里不再赘述。

秘笈心法

心法领悟 602：header()函数的常用功能。

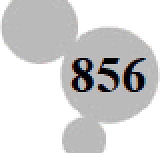
header()函数的常用功能有：

（1）重定向，这是最常用的功能。

```
<?php
    header("Location: http://www.mrbccd.com");
?>
```

（2）强制客户端每次访问页面时获取最新资料，而不是使用存在于客户端的缓存。

```
<?php
//设置页面的过期时间（用格林威治时间表示）
   header("Expires:    Mon,    08    Jul    2008    08:08:08    GMT");
```



```
        //设置页面的最后更新日期（用格林威治时间表示），使浏览器获取最新资料
        header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "GMT");
        header("Cache-Control: no-cache,     must-revalidate");              //控制页面不使用缓存
        header("Pragma: no-cache");                                          //参数与以前的服务器兼容，即兼容 HTTP 1.0 协议
        header("Content-type:     application/file");                        //输出类型
        header("Content-Length:     227685");                                //文件长度
        header("Accept-Ranges:     bytes");                                  //接受的范围单位
        //缺省时文件保存对话框中的文件名称
        header("Content－Disposition: attachment; filename=$filename");     //实现下载
?>
```

（3）输出状态值到浏览器，控制访问权限。

```
<?php
 header('HTTP/1.1 401 Unauthorized');
 header('status: 401 Unauthorized');
?>
```

（4）实现 HTTP 下载。

```
header("Content-type: application/x-gzip");
header("Content-Disposition: attachment; filename=文件名");
header("Content-Description: PHP3 Generated Data"); >
```

HTTP 标头有很多，这里只介绍下载的 HTTP 标头，其代码如下：

```
header('Content-Disposition: attachment; filename="filename"');
```

在应用过程中，唯一需要改动的就是 filename，即将 filename 替换为要下载的文件。

实例 603	图书详细信息展示 光盘位置：光盘\MR\15\603	高级 趣味指数：★★★☆

实例说明

为了让浏览者能够更加细致地了解所查询图书的详细信息，并能够对所选购的图书进行评论，还需要通过图书详细信息页将图书信息展示出来，效果如图 15.30 所示。



图 15.30　图书详细信息展示

关键技术

图书详细信息展示页中应用的技术如下：

（1）数据库的连接、操作类。

```
tb_bookinfo.smalltypeid = tb_smalltype.id and tb_bookinfo.id = '" . $bid . "'", $connID);   //获得图书信息
$smarty->assign('bookinfo', $bookinfo);
```

代码说明：

❶ 这里将用户所浏览过的图书 ID 用字符“@”连接保存到 SESSION 中。

❷ 函数 array_push()用来将元素添加到指定的数组中。

（2）为了帮助读者选购套系图书，制订学习流程，在制作明日科技图书网的图书详细信息页时，增加图书组合购买功能，其效果如图 15.31 所示。



图 15.31　图书组合购买

在数据库中使用 tb_book 表的 bookids 字段将组合购买图书的 ID 用字符“@”连接，在 bookinfo.php 页面中，首先使用函数 explode()将组合 ID 保存到数组中，然后使用 for 循环语句遍历该数组，根据该数组元素查询出所有组合图书信息并将图书信息保存到$arrayZhBooks 数组中，最后将该数组传递给模板，在模板中通过 section 语句输出所有组合图书信息，其关键代码如下：

```
$arrayBookids = explode('@', $bookinfo[0]['bookids']);          //使用字符“@”分隔组合图书 ID
$arrayZhBooks = array();
$zhIDs = $bookinfo[0]['bid'] . '@';
for ($i = 0; $i < count($arrayBookids); $i ++) {               //获得组合 ID 图书信息
    if ($arrayBookids[$i] != '') {
        $tmpBookinfo = $adminDB->executeSQL("select id, bookname, bookimg from tb_bookinfo where id='" . $arrayBookids[$i] . "'", $connID);
        array_push($arrayZhBooks, $tmpBookinfo);
        $zhIDs .= $tmpBookinfo[0]['id'] . '@';
    }
}
$smarty->assign('zhIDs', $zhIDs);
$smarty->assign('arrayZhBooks', $arrayZhBooks);
```

（3）在图书详细信息页加入读者评论功能，有助于网站经营者分析图书的价值，从而确定图书在站点中的摆放位置，实现评论功能的过程如下：

① 在图书详细信息展示的模板页中创建 form 表单，将用户评论提交到 bookinfo.php 页面，添加到指定的数据表中。

② 在 bookinfo.php 页面中读取图书评论信息，并且将评论信息在模板页 bookinfo.phtml 中输出。同时在模板页中为图书评论信息创建“删除”超链接，使评论的发布者可以删除自己的评论。

本实例的完整代码请参考本书光盘，由于篇幅所限这里不再赘述。

秘笈心法

心法领悟 603：内容简介和图书目录之间的切换。

在图书详细信息展示的模板页中，通过 onmouseover 事件调用 JavaScript 脚本中的 changeTab1()方法实现在内容简介和图书目录之间的切换，其原理是通过 onmouseover 事件调用 JavaScript 脚本中的方法，控制<div>标签样式的隐藏和展开。

应用的过程中，唯一需要改动的就是 filename，即将 filename 替换为要下载的文件。

实例 604	新闻公告 光盘位置：光盘\MR\15\604	高级 趣味指数：★★★☆

实例说明

在明日科技图书网的主页中有一个新闻公告展示模块，用于展示明日科技网站发布的新闻公告信息，其运行效果如图 15.32 所示。



图 15.32　新闻公告



在本实例中，运用了两种截取中文字符串的方法。第一种是 Smarty 模板中的 truncate 方法，它虽然可以对中文字符串进行截取，但是同样可能会出现乱码的问题；第二种是自定义的方法 msubstr()，可以防止截取中文字符串时出现乱码，在使用时需要传递 3 个参数，指定截取的字符串、截取的开始位置和截取的长度。

15.5　站 内 搜 索

搜索功能是一个很重要且常见的功能。一个好的搜索引擎能给用户的使用带来方便。本节将以明日科技图书网站中的一般搜索和高级搜索为例，讲解搜索功能的实现方法。

实例 605	一般搜索 光盘位置：光盘\MR\15\605	高级 趣味指数：★★★☆

实例说明

在明日科技图书网站的头文件中，设计了一个查找图书的文本框，其实现的就是一般搜索的功能。在本实例中，按图书名称进行搜索，输入 PHP 关键字，单击“查找图书”按钮，将输出如图 15.33 所示的内容。



图 15.33　查找图书信息

关键技术

在一般搜索功能中，关键技术是应用 explode()函数对查询的关键字进行拆分返回一个数组。然后通过 for 循环语句读取这个数组中的元素。最后定义 select 查询中的 where 子句，以数组中的元素值为 like 模糊查询的条件，其关键代码如下：

```
$arrayKeyWord = explode(' ', $keyWord);                    //使用空格对查询的关键字进行拆分
$where = "";                                               //定义变量
$j = 0;
for($i=0; $i<count($arrayKeyWord); $i++){                  //循环读取查询关键字数组中的元素
    if($arrayKeyWord[$i]!=''){                             //判断数组元素的值是否为空
        if($j>0){                                          //如果变量值大于 0
            $where .= " or ";                              //定义 where 变量的值
        }
        $where .= " tb_bookinfo.bookname like '%".$arrayKeyWord[$i]."%' ";    //定义变量的值，即定义查询语句的 where 子句
        $j++;
    }
}
```

如果参数 limit 是负数，则返回除最后的−limit 个元素外的所有元素。

实例 606　**高级搜索**　光盘位置：光盘\MR\15\606　高级　趣味指数：★★★★☆

实例说明

在明日科技图书网站的高级搜索功能中，可以输入图书名称关键字、作者、出版社、类别和出版时间，根据这些内容进行综合查询，高级搜索的运行效果如图 15.34 所示。

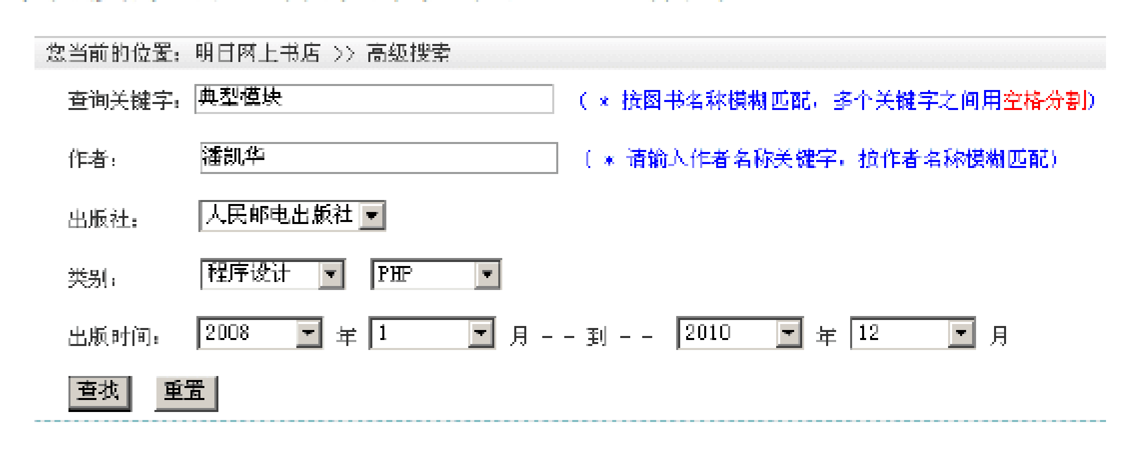


图 15.34　高级搜索表单

关键技术

（1）应用 Smarty 模板中的{ldelim}{rdelim}标签在 Smarty 模板页中直接编辑 JavaScript 脚本语句。

（2）在 JavaScript 脚本中应用 jQuery 技术编写图书类别展示的二级联动菜单，其关键代码如下：

```
<script src="{util->baseUrl}/js/jquery.js"></script>            <!--载入 jQuery 库-->
<script language="javascript" >
$(document).ready(
      function(){ldelim}
        $("#bigtypeid").change(function(){ldelim}
                $.get("{util->baseUrl}/getstype.php?stid="+$("#bigtypeid").val(), null, function(data){ldelim}
                      $("#smalltypeid").empty();
                      $("#smalltypeid").append(data);
              {rdelim});
          {rdelim});
      {rdelim}
);
</script>
```

其中调用 getstype.php 文件，定义图书类别中子类的下拉列表框值，getstype.php 文件的代码如下：

```
<?php
require_once 'lzh.inc.php';                    //包含配置文件
$smalltypes = $adminDB->executeSQL("select id, typename from tb_smalltype where bigtypeid='".$_GET['stid']."'", $connID);
$data = "";
for($i = 0; $i < count($smalltypes); $i++){     //将从数据库中读取的数据添加到下拉列表的选项中
     $data .= "<option value=\"".$smalltypes[$i]['id']."\">".iconv('gbk', 'utf-8', $smalltypes[$i]['typename'])."</option>";
}
echo $data;                                     //输出变量值，即输出下拉列表框的值
```

（3）应用 Smarty 模板中的 html_options 方法自定义函数设置日期的下拉列表框值，其部分代码如下：

```
<select name="fyear">
<option value='' selected>-请选择-</option>
      {html_options options = $arrayYear}
</select> 年
<select name="fmonth">
     <option value='' selected>-请选择-</option>
      {html_options options = $arrayMonth}
</select> 月 - - 到
```

（4）定义高级搜索的 SQL 语句，充分发挥 where 子句的作用，通过多表查询完成高级搜索的功能。

（4）在 search.phtml 模板页中应用 section 语句分页输出查询结果，代码请参考本书光盘。

秘笈心法

心法领悟 606：jQuery 框架。

jQuery 由美国人 John Resig 创建，至今已吸引了来自世界各地的众多 JavaScript 高手加入其团队。jQuery 是继 prototype 之后又一个优秀的 JavaScript 框架，其宗旨是 WRITE LESS，DO MORE，即写更少的代码，做更多的事情。它是轻量级的 js 库（压缩后只有 21KB），这是其他 js 库所不及的。它兼容 CSS3 以及各种浏览器（IE 6.0+、FF 1.5+、Safari 2.0+、Opera 9.0+等）。

jQuery 是一个快速的、简洁的 JavaScript 库，使用户能更方便地处理 HTML documents、events、实现动画效果，并且方便地为网站提供 AJAX 交互。

应用 jQuery 技术的前提条件是必须在项目中载入 jQuery 库，否则无法调用其中的方法。

15.6　购　物　车

购物车是网上购物时使用的一个临时存储商品的“车辆”，为用户在网上购物提供了很大的方便，不用担心一次购买多个商品时要进行多次提交结算的操作，可以将所选商品放入购物车中，等选购完成后一起结算，购物车是电子商务类网站中一个必不可少的功能。本节将以明日网上书店中的购物车为模板，详细讲解购物车的实现过程。

实例 607	购物车类 光盘位置：光盘\MR\15\607	高级 趣味指数：★★★☆

实例说明

对于电子商务类网站来说，安全、稳定的购物车模块是非常重要的，该模块直接关系到交易是否能够顺利进行。为了提高代码的可移植性，在制作明日科技书店购物车时，将购物车抽象成类，封装到 library\Cart.php 文件中。购物车功能的运行效果如图 15.35 所示。



图 15.35　购物车的运行效果

关键技术

在购物车类的封装中，应用最多的就是 explode()函数和 implode()函数。

（1）explode()函数将字符串用指定的子串分隔，并将分隔结果保存到数组中。在购物车类的 addCart()方法中，应用 explode()函数将 ID 串用“@”分隔，并将分隔结果保存到数组中，其关键代码如下：

```
$arrayIds = explode('@', $this->idStr);                    //将 ID 串用“@”分隔，并将分隔结果保存到数组中
```

（2）implode()函数将指定数组中的所有元素使用特定的字符进行连接形成新的字符串，其语法如下：

商品 ID 相同的 ID 号，则使用函数 unset()将数组元素销毁，然后再将数组中所有元素重新用字符“@”连接成商品 ID 串，这样就可以实现将指定的商品从购物车中移除的操作。

（5）使用 changeNum()方法更改购物车中商品的数量，其代码如下：

```
function changeNum($goodsId, $goodsNum){
    $arrayIds = explode('@', $this->idStr);                //将商品 ID 串用“@”分隔成数组
    $arrayNums = explode('@', $this->numStr);              //将商品数量串用“@”分隔成数组
    for($i = 0; $i < count($arrayIds); $i++){              //遍历商品 ID 串
        if ($arrayIds[$i] == $goodsId){                    //如果查找到要更改的 ID
                $arrayNums[$i] = $goodsNum;                //更改数量
        }
    }
  $this->idStr = implode('@', $arrayIds);                  //重新合并成 ID 串
    $this->numStr = implode('@', $arrayNums);              //重新合并成数量串
}
```

changeNum()方法中包含两个参数，分别为要更改的商品 ID 号和要更改的数量。在该方法内部，首先使用函数 explode()将商品 ID 串和商品数量串用“@”进行分隔，并将分隔的字串分别保存到数组中，然后通过 for 循环语句遍历商品 ID 数组，在其中查找要更改数量的商品 ID，查找到后，将新数量值赋给数量串中对应的数组元素，最后使用函数 implode()将商品 ID 数组和商品数量数组中的所有元素分别重新用字符“@”进行连接，组成新的商品 ID 串和商品数量串。

（6）通过 Cart 类的 setCartNull()方法，清空购物车中的商品。实现清空购物车中商品的操作，只需将商品 ID 串和商品数量串赋为空值即可，其代码如下：

```
function setCartNull(){
    $this->idStr = '';                                     //清空商品 ID 串
    $this->numStr = '';                                    //清空商品数量串
}
```

（7）通过购物车类中的方法完成对商品 ID 串和商品数量串的操作后，还需要定义两个方法，分别用来获取商品 ID 串和商品数量串，其代码如下：

```
function getIdStr(){                                       //获得购物车的商品 ID 串
    return $this->idStr;
}
function getNumStr(){                                      //获得购物车的商品数量串
    return $this->numStr;
}
```

以上讲解了明日科技书店购物车功能的 Cart 类的属性和主要方法，其存储于 library\Cart.php 文件中。

秘笈心法

心法领悟 607：解析购物车类中的商品 ID 串和数量串。

购物车类中的商品 ID 串和商品数量串是指用字符“@”连接的多个商品的 ID 号和数量值，其目的是使用一个变量或者数据表中的一个字段存储多个商品的 ID 或数量。

实例 608	购物车功能实现 光盘位置：光盘\MR\15\608	高级 趣味指数：★★★☆

实例说明

购物车的主要功能是保留用户选择的商品信息。用户可以在购物车内设置选购商品的数量、显示选购商品的总金额，还可以进行清除选择的全部商品信息、重新选择商品信息等操作。在实例 607 中已经讲解了购物车类的实现过程，本实例将在此基础上介绍购物车的页面设计和功能实现，其中明日科技书店的购物车运行效果如图 15.36 所示。

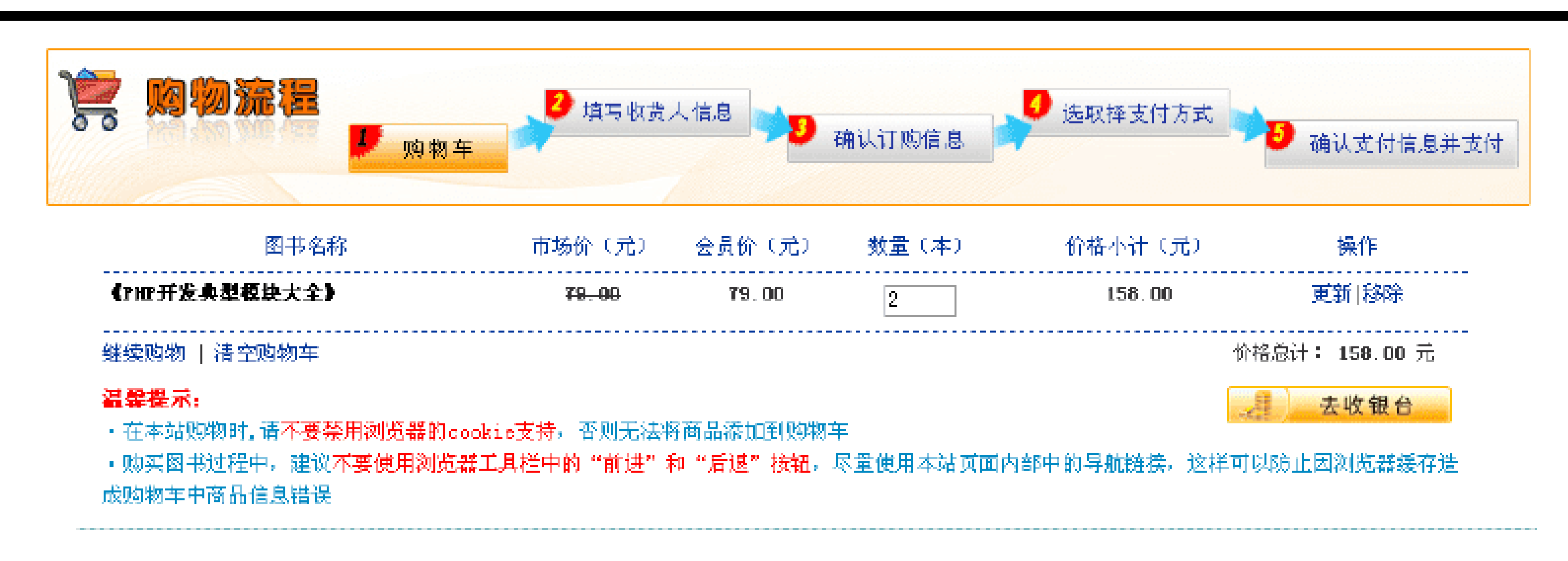


图 15.36　明日科技书店购物车运行效果

关键技术

购物车功能实现的前提是为每位用户发放一辆购物车，即为每位用户定义一个商品 ID 串和商品数量串，并将其保存在$_SESSION 全局数组中。

所以在购物车功能实现的 cart.php 文件中，首先要初始化 SESSION 变量，判断是否已经设置了用于保存商品 ID 串的$_SESSION['idStr']数组元素，以及保存商品数量串的$_SESSION['numStr']数组元素。如果没有设定，说明是第一次将商品添加到购物车中，则设置这两个数组元素的值为空。然后对购物车类 Cart 进行实例化，以便继续对购物车中的商品进行增、删、改操作，其关键代码如下：

```
require_once 'header.php';                    //包含网站头文件以及配置文件，在配置文件中初始化 SESSION 变量
if (! isset($_SESSION['idStr']) || ! isset($_SESSION['numStr'])) {
    $_SESSION['idStr'] = '';
    $_SESSION['numStr'] = '';
}
$cart = new Cart($_SESSION['idStr'], $_SESSION['numStr']);
```

这是购物车功能实现的前提条件，只有满足了上述条件，才能够继续执行购物车中的增、删、改等功能。

设计过程

下面在购物车类的基础上讲解购物车各个功能的实现过程，其操作在 cart.php 文件中进行。

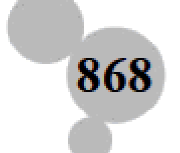
（1）有关购物车的分配已经在关键技术中讲解，这里不再赘述。

（2）应用购物车类的 addCart()方法将商品添加到购物车，其传递的参数值分别为购买商品的 ID 和购买商品的数量，将商品成功添加到购物车后，还需要使用购物车类的 getIdStr()和 getNumStr()方法分页获取经购物车类处理后的商品 ID 串和商品数量串，将获取结果分别赋给$_SESSION 全局数组中对应保存商品 ID 串和商品数量串的数组元素，实现购物车中商品刷新效果，商品添加的代码如下：

```
if($t == 'add') {                                    //判断操作类型
    $cart->addCart($_GET['bid'], 1);                 //调用购物车类的 addCart()方法将商品添加到购物车中
    $_SESSION['idStr'] = $cart->getIdStr();          //获取商品 ID 串
    $_SESSION['numStr'] = $cart->getNumStr();        //获取商品数量串
}
```

（3）在实例 607 中已经介绍过，可以将与某本图书相关的所有图书批量添加到购物车中，其原理是：首先，通过 explode()函数将保存组合图书 ID 的字符串用字符“@”分隔，并将分隔结果保存到数组中，这样该数组的每个元素为各本组合图书的 ID 号。然后，通过 for 循环遍历数组，在遍历过程中分别通过购物车类的 addCart()方法将图书添加到购物车中，其具体实现代码如下：

```
if($t == 'gAdd'){                                    //判断操作类型，如果为 gAdd 表示组合添加
    $arrayGIds = explode('@', $_GET['ids']);         //分隔组合 ID 并将结果保存到数组中
    for($i=0; $i<count($arrayGIds); $i++){           //遍历数组
        if($arrayGIds[$i]!='' && !in_array($arrayGIds[$i], explode('@', $_SESSION['idStr']))){
            $cart->addCart($arrayGIds[$i] ,1);       //分别将每本图书添加到购物车中
        }
    }
    $_SESSION['idStr'] = $cart->getIdStr();          //重新为商品 ID 串的 SESSION 变量赋值
    $_SESSION['numStr'] = $cart->getNumStr();        //重新为商品数量串的 SESSION 变量赋值
}
```



```
        $_SESSION['toUrl'] = 'getbuyuserinfo.html';
}
```

（8）在 views 文件夹下新建 cart.phtml 模板页，完成购物车页面的设计。

秘笈心法

心法领悟 608：判断指定的数据是否为正整数。

判断指定的数据是否为正整数，应用到 3 个函数：is_numeric()、intval()和 floor()。

is_numeric()函数检测变量是否为数字或者数字字符串，如果是则返回 TRUE，否则返回 FALSE。

intval()函数获取变量的整数值。

floor()函数利用舍去法获取变量的整数值。

在 if 条件语句中，判断 is_numeric()函数的返回值是否为 TRUE，判断 intval()函数的返回值是否大于 0，判断 floor()函数的返回值与变量值是否相等。当满足 is_numeric()函数的返回值为 TRUE，intval()函数的返回值大于 0，floor()函数的返回值与变量值相等时，说明变量是正整数，否则说明不是正整数，其应用方法如下：

```
if (! is_numeric($changenum) || intval($changenum) <= 0 || floor($changenum) != $changenum) {
        echo "<script>alert('购买图书的数量只能为正整数！');</script>";
}
```

其中变量$changenum 为要验证的变量。

15.7　图书订单处理

为了方便用户购买图书促进图书销售，在开发明日科技图书网时特别设立图书订单处理模块，通过该模块用户可以方便地订购图书并进行在线支付。本节将对明日科技书店图书订单处理模块进行讲解。

实例 609	填写收货人信息 光盘位置：光盘\MR\15\609	高级 趣味指数：★★★☆

实例说明

为了能够将用户所购买的图书顺利邮寄到用户手中，在用户选择好购买的图书后还需要填写收货人信息，明日科技图书网用于填写收货人信息的页面如图 15.37 所示，在该页面中，用户需要填写个人的真实联系信息、收货方式等内容，如果需要发票，还需要填写发票抬头。

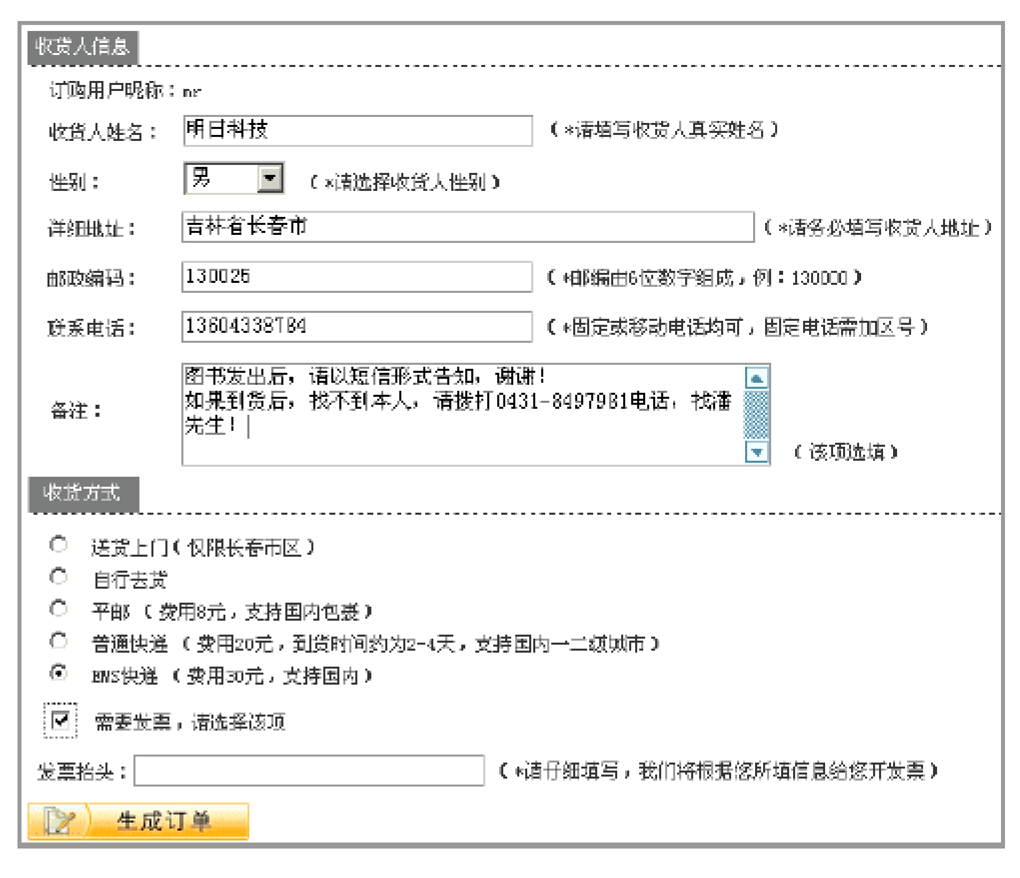


图 15.37　填写收货人信息的页面

```
            $recpay = '20';
        } elseif ($rect == '4') {
            $recpay = '30';
        }
        $_SESSION['recuserinfo']['recpay'] = $recpay;
        echo "<script>window.location.href='" . $util->baseUrl() . "/deforder.html';</script>";                    //重新定向到 deforder.html 页面
        exit();
}
$smarty->display('getbuyuserinfo.phtml');
require_once 'footer.php';
```

秘笈心法

心法领悟 609：在 Smarty 模板中直接获取 SESSION 变量的值。

在 Smarty 模板中直接获取 SESSION 变量的值，应用的是 Smarty 模板中的$smarty.session 标签。首先在 PHP 动态页中定义 SESSION 变量，其格式如下：

```
$_SESSION['recuserinfo']['username'] = $_POST['username'];
$_SESSION['recuserinfo']['sex'] = $_POST['sex'];
```

然后在 Smarty 模板页中，通过如下方法获取 SESSION 变量的值：

```
$smarty.session.recuserinfo.username
$smarty.session.recuserinfo.sex
```

实例 610 确认订购信息

光盘位置：光盘\MR\15\610

高级

趣味指数：★★★★☆

实例说明

用户填写完收货人信息后将进入订购信息确认页面，如图 15.38 所示，在该页面中可以查看并确认用户所订购图书的名称、单价、总价格和收货人的信息。



图 15.38　确认订购信息

关键技术

确认订购信息功能的实现，首先应用的是 explode()函数对商品 ID 串和商品数量串中的数据进行读取，然后根据返回的数组元素值从数据库中读取图书信息，完成图书价格的汇总。

在 deforder.phtml 模板页中，通过 section 语句循环输出用户购买的图书信息，并且通过$smarty.session 标签输出收货人的信息。上述方法都已经在前面的实例中讲解过，这里不再赘述。

设计过程

（1）为了使图书订购流程的业务逻辑更加方便，将购物车和收货人的信息保存到 SESSION 中，所以显示



超链接，将跳转到如图 15.39 所示的页面，输出本次图书订购的订单号、支付方式和需要支付的总金额。

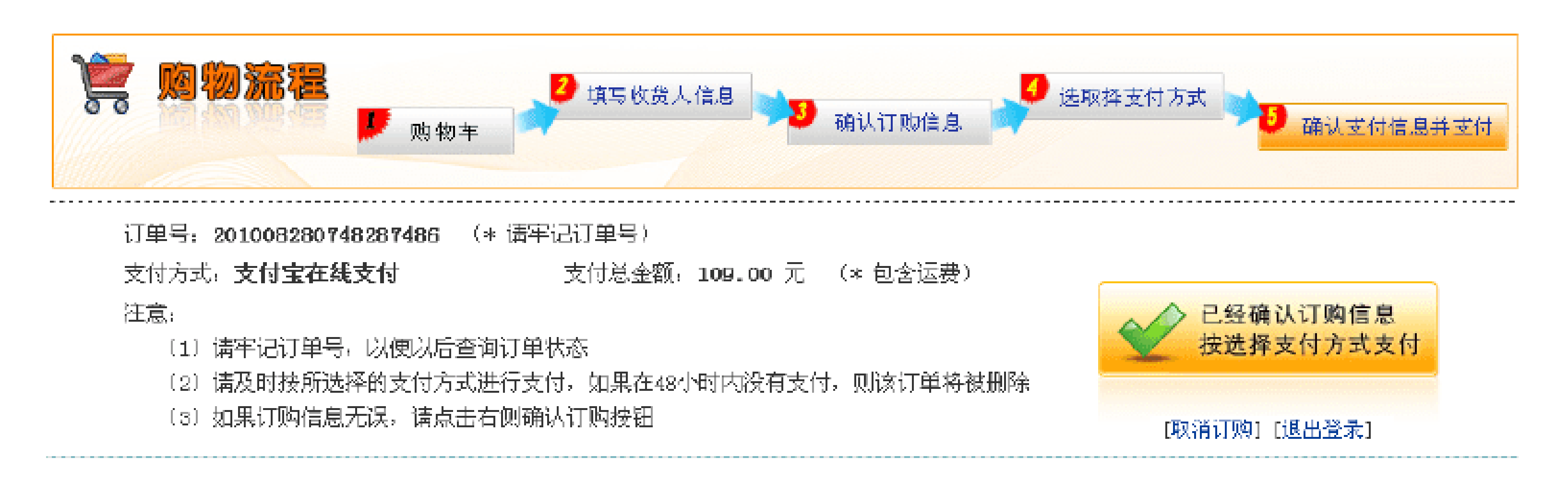


图 15.39　支付宝在线支付

如果单击图 15.39 中的“已经确认订购信息 按选择支付方式支付”按钮，将跳转到支付宝页面完成支付操作。

关键技术

使用支付宝进行在线支付，必须先在支付宝中注册一个账户，然后下载支付宝提供的接口文件，实现电子商务网站与支付宝的交互。

在电子商务网站中，需要嵌入支付宝提供的几个文件，如图 15.40 所示。

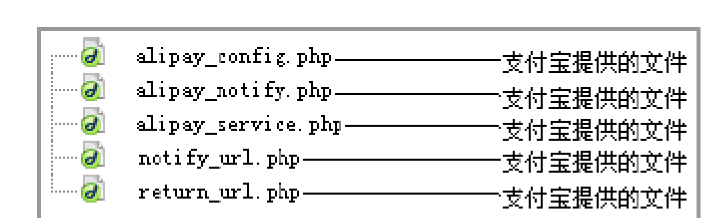


图 15.40　支付宝提供的文件

在支付宝提供的接口文件中，提供了完整的实例程序，只要依照实例程序中的操作方法，将电子商务网站中指定的数据提交到支付宝提供的数组参数中，支付宝即会通过自己定义的方法将数据提交到指定的网站中。

设计过程

（1）单击确认订购信息页面中的“选择支付方式”按钮，进入如图 15.41 所示的支付方式选择页面。



图 15.41　支付方式选择页面

新建 selectpaytype.php 文件，包含网页的头文件和尾文件，并指定模板页 selectpaytype.phtml；新建 selectpaytype.phtml 模板页，输出各种支付方式并且设置进入指定支付方式的超链接。

（2）单击支付宝支付的“支付”超链接，将跳转到确认支付方式页面，该页面由 defpay.php 和 defpay.phtml 组成。

在 defpay.php 动态页中，根据超链接传递的参数值确定选择的支付方式和类型，并且将存储在 SESSION 变量中的订单信息添加到数据表中。同时判断如果选择的是支付宝在线支付（$alertType = '2'），则定义向支付宝

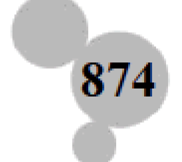




图 15.42　支付宝在线支付



图 15.43　工行在线支付

关键技术

工行在线支付功能的实现主要根据工行提供的在线支付的接口，将订单中的数据以表单的形式提交到工行指定的网址中。详细的参数设置可以参考工行提供的文档说明书和实例程序。

在实现与工行接口的连接之前，首先要从工行提供的说明书中找到两个文件：开发 API 接口\LIB\windows\win32\infosecapi.dll（XP 系统使用的文件）和开发 API 接口\ASP\ ICBCEBankUtil.dll，并将其复制到系统文件夹下的 Windows\system32 下，在运行命令中输入 regsvr32 ICBCEBankUtil.dll，加载服务。

通过 PHP 实现网站与工行接口交互的方法如下：

首先，通过 com 类中的 init()方法对订单签名数据和商城的公钥进行加密处理，并且将指定的数据以表单的形式提交到工行指定的页面中，其关键代码如下：

```
$src="ICBC_PERBANK_B2C1.0.0.0".$infomer["merid"].$infomer["meracct"].$returnaddress."HS".$ddnumber.$amount."0010".$nowtime."0";
$com=new com('ICBCEBANKUTIL.B2CUtil');
$rc=$com->init("c:\WINDOWS\user.crt","d: \user.crt","d: \user.key","00");
$ssrc=$com->signC($src, strlen($src));                          //订单签名数据
$rc=$com->verifySignC($src,strlen($src), $ssrc, strlen($ssrc));
$cert=$com->getCert(1);                                          //商城证书公钥
```

完成订单签名数据和商城证书公钥的加密处理后，将商户的测试证书存储到 C:\Windows 目录下，设置商户的私钥口令为 00。

然后创建 form 表单，将指定的数据提交到工行指定的网站中。

设计过程

工行在线支付的操作同样由 defpay.php 和 defpay.phtml 两个文件组成。

（1）在 defpay.php 动态页中，判断如果选择的是工行在线支付（$alertType = '3'），则定义向工行在线支付网站中提交的数据，并且将数据赋给模板变量，在模板页中通过 form 表单将数据提交到工行在线支付的网站中。在 defpay.php 动态页中，有关工行在线支付的关键代码如下：

```
<?php
require_once 'header.php';
$t = $_GET['t'];
if ($t == 'qyzh') {
      $paytype = '企业账户汇款';
      $alertType = '0';
}elseif ($t == 'zfb') {
      $paytype = '支付宝在线支付';
      $alertType = '2';
} elseif ($t == 'ghzxzf') {
      $paytype = '工行在线支付';
      $alertType = '3';
}
//省略了部分代码
if ($alertType == '3') {
      $merInfo = $adminDB->executeSQL("select merid, meracct from tb_system where mark = 1", $connID);
      $merid = $merInfo[0]['merid'];
      $meracct = $merInfo[0]['meracct'];
      $merURL = 'http://www.mrbooks.cn';
```



图 15.44　后台管理系统登录

关键技术

在后台管理系统登录功能实现过程中，应用 JavaScript 脚本调用 xym.php 文件完成验证码的输出操作。在 xym.php 文件中，应用文件系统函数库中的函数将指定的图片以二进制的形式输出到浏览器，其代码如下：

```
<?php
$num=$_GET[num];                              //获得传递的数字
$address="img/code/".$num.".gif";             //获取对应图片地址
$fp=fopen($address,"r");                      //打开对应的图片
echo fread($fp,filesize($address));           //将图片以二进制的形式输出到浏览器
fclose($fp);                                  //关闭打开的图片
?>
```

有关文件系统函数的应用可以参考本书第 4 章的内容，这里不再赘述。

设计过程

明日科技图书网后台登录页面的制作过程主要包括登录表单的设计和登录信息验证。

（1）后台用户登录表单的 UI 是通过 HTML 语言的<input>标签实现的，并通过 JavaScript 实现图片验证码，admin-login.phtml 的关键代码如下：

```
<form name="form_login" method="post" action="{util->baseUrl}/admin-login.php" onsubmit="return chkinputadmin(this)">
<div style="width:300px; height:200px; text-align:left">
用户名：<input type="text" name="anc" size="22" class="input" /><br /><br />
密码：<input type="password" name="pwd" size="22" class="input" /><br /><br />
验证码：<input type="text" name="xym" size="8" class="input" /><input type="hidden" value="" name="xym1">
<script language="javascript">
    var num1=Math.round(Math.random()*10000000);
    var num=num1.toString().substr(0,4);
    document.write("<img name=codeimg4 src='xym.php?num="+num.substr(0,1)+"'>");
    document.write("<img name=codeimg5 src='xym.php?num="+num.substr(1,1)+"'>");
    document.write("<img name=codeimg6 src='xym.php?num="+num.substr(2,1)+"'>");
    document.write("<img name=codeimg7 src='xym.php?num="+num.substr(3,1)+"'>");
    form_login.xym1.value=num;
    function code_1(){ldelim}
        var num1=Math.round(Math.random()*10000000);
        var num=num1.toString().substr(0,4);
        document.codeimg4.src="xym.php?num="+num.substr(0,1);
        document.codeimg5.src="xym.php?num="+num.substr(1,1);
        document.codeimg6.src="xym.php?num="+num.substr(2,1);
        document.codeimg7.src="xym.php?num="+num.substr(3,1);
        form_login.xym1.value=num;
        {rdelim}
</script>
<a href="javascript:code_1()" class="a1">看不清</a>
<input type="submit" value="登录" />  <input type="reset" value="重置" />
</div>
</form>
```

```
<noscript><iframe src="*.html"></iframe></noscript>
<frameset cols="182, *">
 <frame src="{util->baseUrl}/admin-left.php" name="leftwindow">
 <frame src="{util->baseUrl}/admin-main.php" name="mainwindow">
</frameset><noframes></noframes>
```

■ 设计过程

后台管理系统的主页主要由两个文件组成：admin-index.php 和 admin-index.phtml。

（1）在 admin-index.php 文件中，首先包含系统的配置文件。然后判断当前用户是否具有管理员的权限，如果不具备则给出“禁止非法登录”的提示信息，并跳转到网站的首页。最后指定后台管理系统主页的模板页 admin-index.phtml，其代码如下：

```
<?php
require_once 'lzh.inc.php';
if(!isset($_SESSION['anc']) || $_SESSION['anc'] == ''){
      echo "<script>alert('禁止非法登录！');window.location.href='".$util->baseUrl()."/index.html';</script>";
      exit();
}
$smarty->display('admin-index.phtml');
```

（2）新建 admin-index.phtml 模板页，应用 frame 框架完成后台管理系统主页的设计，其代码如下：

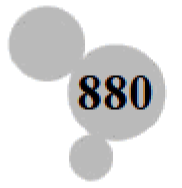
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>明日网上书店管理中心</title>
<link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<noscript><iframe src="*.html"></iframe></noscript>
<frameset cols="182, *">
 <frame src="{util->baseUrl}/admin-left.php" name="leftwindow">
 <frame src="{util->baseUrl}/admin-main.php" name="mainwindow">
</frameset><noframes></noframes>
```

（3）左侧导航页面由一个动态 PHP 文件 admin-left.php 和一个模板页 admin-left.phtml 组成。在 admin-left.php 文件中包含系统配置文件，指定模板页。

在模板页 admin-left.phtml 中，应用 jQuery 技术设计后台管理系统的导航菜单，通过超链接中的 target 属性指定链接文件在 mainwindow 框架中输出，其关键代码如下：

```
<head>
<title>明日网上书店管理中心 - 左侧导航</title>
      <meta http-equiv="Content-Type" content="text/html; charset="gbk" />
      <link rel="stylesheet" type="text/css" href="{util->baseUrl}/css/style.css" />
</head>
<script src="{util->baseUrl}/js/jquery.js"></script>
<script src="{util->baseUrl}/js/adminleft.js"></script>
<div style="width:170px; border:1px solid #FFFFFF">
      <div id="adminsystembutton" style="width:170px; height:34px;    background-color:#4995A8; cursor:hand; background:url({util->baseUrl}/img/
button_xtgl.gif)">
      </div>
      <div id="adminsystem" style="width:100%px; height:25px; padding-top:5px; font-size:13px; background-color:#A3C7D0; line-height:20px;
display:none">
      <li style="display:inline; width:100%; text-align:left; padding-left:30px">|- <a href="{util->baseUrl}/admin-default.php" class="a1"
target="mainwindow">系统信息设置</a></li>
      <li style="display:inline; width:100%; text-align:left; padding-left:30px">|- <a href="{util->baseUrl}/admin-changepwd.php" class="a1"
target="mainwindow">更改管理员密码</a></li>
      <li style="display:inline; width:100%; text-align:left; padding-left:30px">|- <a href="{util->baseUrl}/admin-logout.php" class="a1"
target="_parent">注销退出</a></li>
      </div>
</div>
```

（4）右侧页面默认展示由 admin-main.php 和 admin-main.phtml 文件组成的内容。在 admin-main.php 文件中，首先包含后台管理系统的头文件 admin-header.php，然后指定模板页 admin-main.phtml，最后包含尾文件 admin-footer.php。在 admin-main.phtml 模板页中，输出后台管理系统的系统信息。



秘笈心法

心法领悟 614：超链接中 target 属性的应用。

超链接中 target 属性包含 5 个属性值：_blank 在新窗口中打开超链接文件；_parent 将链接文件载入含有链接框架的父框架或窗口中，如果含有该链接的框架不是嵌套的，则在浏览器全屏窗口中载入链接的文件，与_self 参数相同；_self 在同一框架或窗口中打开链接的文件，此参数为默认值，通常不必指定；_top 在当前的整个浏览器窗口中打开链接的文件，因此会删除所有框架；name 在指定的窗口或框架中打开网页。

在本实例的 admin-left.phtml 模板页中，通过超链接的 target 属性指定 mainwindow 框架，在该框架中打开链接的网页。

15.9　系统管理

系统管理主要用于对系统公用信息进行设置及对管理员密码进行更改，通过该模块可以提高系统的可维护性。

实例 615	系统信息设置 光盘位置：光盘\MR\15\615	高级 趣味指数：★★★☆

实例说明

明日科技图书网的系统信息设置主要用于设置商城 ID、企业账号、企业版权、联系方式和程序文件的保存路径等信息，系统信息设置的页面如图 15.46 所示。



图 15.46　系统信息设置页面

关键技术

在本实例中，关键之处不在于系统信息设置功能的实现方法，更重要的是它的操作。在进行系统信息设置时，必须设置最后 3 项内容，它们指定的是系统中图片、广告和试读文件的下载地址，如果设置的路径不正确，那么这 3 项内容将不能正确输出。

笔者在编写此实例时，将其路径设置为 http://127.0.0.1/MR/15/005/，这是在本地服务器运行的路径，如果将这个程序上传到互联网的服务器中，就必须修改这个路径，否则将找不到指定的图片和文件。

设计过程

系统信息设置功能由 admin-default.php 和 admin-default.phtml 两个文件组成。

（1）在 admin-default.php 文件中，完成数据库中存储的系统信息的输出和更新两项操作。首先包含后台管理系统的头文件，然后判断 form 表单中提交的 merid 值是否为空，如果不为空，则应用数据库管理类的 executeSQL()方法执行更新语句，更新数据库中存储的系统信息。最后应用数据库管理类的 executeSQL()方法执行查询操作，查询数据库中存储的系统信息并将其存储到模板变量中，其代码如下：

```
<?php
require_once 'admin-header.php';                                   //包含网站头文件
if (isset($_POST['merid']) && $_POST['merid'] != '') {             //判断表单提交的数据是否为空
    if (! $adminDB->executeSQL("update tb_system set merid='" . $_POST['merid'] . "', meracct='" . $_POST['meracct'] . "', readurl='" .
$_POST['readurl'] . "', bookimgurl='" . $_POST['bookimgurl'] . "', ggurl='" . $_POST['ggurl'] . "', bq='" . $_POST['bq'] . "' , address='" .
$_POST['address'] . "', tel='" . $_POST['tel'] . "', cz='" . $_POST['cz'] . "', email='" . $_POST['email'] . "', icp='" . $_POST['icp'] . "', qq='" . $_POST['qq'] .
"' where mark=1", $connID)) {
        echo "<script>alert('系统信息设置失败！');</script>";
    } else {
        echo "<script>alert('系统信息设置成功！');</script>";
    }
}
$system = $adminDB->executeSQL("select id, merid, meracct, readurl, bookimgurl, ggurl, bq, address, tel, cz, email, icp, qq from tb_system where
mark=1", $connID);
$smarty->assign('system', $system);                                //将查询结果赋给模板变量
$smarty->display('admin-default.phtml');                           //指定模板页
require_once 'admin-footer.php';                                   //包含网站尾文件
```

（2）新建 admin-default.phtml 模板页，创建 form 表单，提交需要更新的系统信息，并且将模板变量中传递的系统信息作为表单元素的默认值。

秘笈心法

心法领悟 615：通过 JavaScript 脚本中的 chkinputsysteminfo()方法验证表单元素值。

在本实例的 admin-default.phtml 模板页中，调用 js/adminfun.js 文件中的 chkinputsysteminfo()方法对系统信息设置表单中提交的数据进行验证。

实例 616	更改管理员密码 光盘位置：光盘\MR\15\616	高级 趣味指数：★★★☆

实例说明

为了便于网站管理者对网站进行管理提高网站的安全性，在系统管理模块增加管理员密码修改功能，如图 15.47 所示。

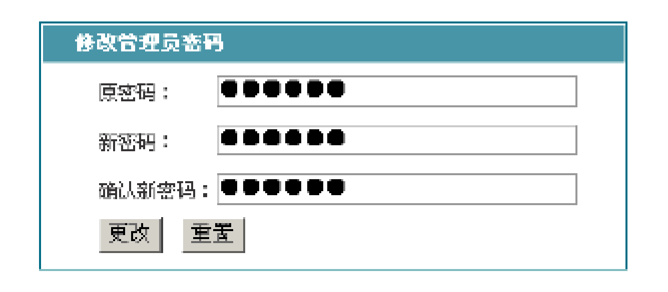
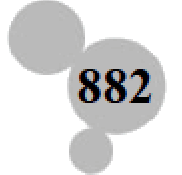


图 15.47　管理员密码修改

关键技术

更改管理员密码的原理：用户在上述表单中输入原密码和新设定的密码，单击“更改”按钮将表单中的信息提交到 admin-changepwd.php 文件中对密码进行更改，在该文件中，首先判断用户输入的原密码是否正确，如



果正确则使用数据库管理类的 executeSQL()方法执行 update 语句对管理员密码进行更改，反之则给出原密码错误的提示信息。

设计过程

后台管理员密码修改功能由 admin-changepwd.phtml 和 admin-changepwd.php 两个文件组成。

（1）新建 admin-changepwd.phtml 模板页，创建 form 表单，将要重新设置的密码提交到 admin-changepwd.php 文件中。

（2）新建 admin-changepwd.php 文件，完成密码更新操作。首先，验证表单中提交的原始密码是否正确，如果不正确则给出提示信息；如果正确，执行密码的更新操作，其代码如下：

```
<?php
require_once 'admin-header.php';                                    //包含网站头文件
if (isset($_POST['oldpwd']) && $_POST['oldpwd'] != '') {            //判断用户提交的原始密码是否存在
    if (! $adminDB->executeSQL("select id, userne from tb_user where userne='" . $_SESSION['anc'] . "' and pwd='" . md5($_POST['oldpwd']) . "'",
$connID)) {                                                         //验证用户提交的原始密码是否正确
        echo "<script>alert('原密码输入有误！');</script>";
    } else {
        if ($adminDB->executeSQL("update tb_user set pwd='" . md5($_POST['pwd1']) . "', truepwd='" . $_POST['pwd1'] . "' where userne='" .
$_SESSION['anc'] . "'", $connID)) {                                 //如果原始密码正确，则执行密码更新操作
            echo "<script>alert('密码更改成功！');</script>";
        } else {
            echo "<script>alert('密码更改失败！');</script>";
        }
    }
}
$smarty->display('admin-changepwd.phtml');                          //指定模板页
require_once 'admin-footer.php';                                    //包含网站尾文件
```

秘笈心法

心法领悟 616：通过 chkChangePwd ()方法验证表单提交的信息。

在本实例的 admin-changepwd.phtml 模板页中，调用 js/adminfun.js 文件中的 chkChangePwd ()方法验证表单中提交的新密码与确认密码是否相同，并验证提交的密码长度是否大于 6 个字节。

15.10　图书类别管理

为了便于用户选购和查看图书信息，在明日科技图书网的图书展示模块实现按图书分类展示和按出版社分类展示功能。本节将介绍如何在网站后台对图书类别进行管理。

实例 617	图书大类管理 光盘位置：光盘\MR\15\617	高级 趣味指数：★★★★

实例说明

在明日科技图书网中，为提高图书分类层次，在图书分类展示模块首先要求用户选择图书大类，然后再选择图书小类来展示图书信息列表。图书大类管理主要包括对类别的添加、修改和删除操作。图书大类管理中浏览图书大类信息的页面效果如图 15.48 所示。

图 15.48　浏览图书大类信息

```
$bigtypes = $adminDB->executeSQL("select id, typename, addtime from tb_bigtype order by addtime ", $connID);    //循环读取数据
$smarty->assign('bigtypes', $bigtypes);                          //将查询结果赋给模板变量
$smarty->display('admin-listbigtype.phtml');                     //指定模板页
require_once 'admin-footer.php';
```

秘笈心法

心法领悟 617：section 语句的应用。

在输出图书大类信息时，应用的是 Smarty 模板中的 section 语句，并且调用注册模板对象中的 unHtml()方法对输出的数据进行格式化的操作。

实例 618　**图书小类管理**　光盘位置：光盘\MR\15\618　**高级**　趣味指数：★★★★☆

实例说明

在图书大类管理的基础上为了更加精确地定位图书，方便读者查找，明日科技图书网的图书展示模块又精确到小类，例如，程序类图书又精确到具体的开发语言。在图书小类中添加小类名称和选择所属大类的运行效果如图 15.49 所示。

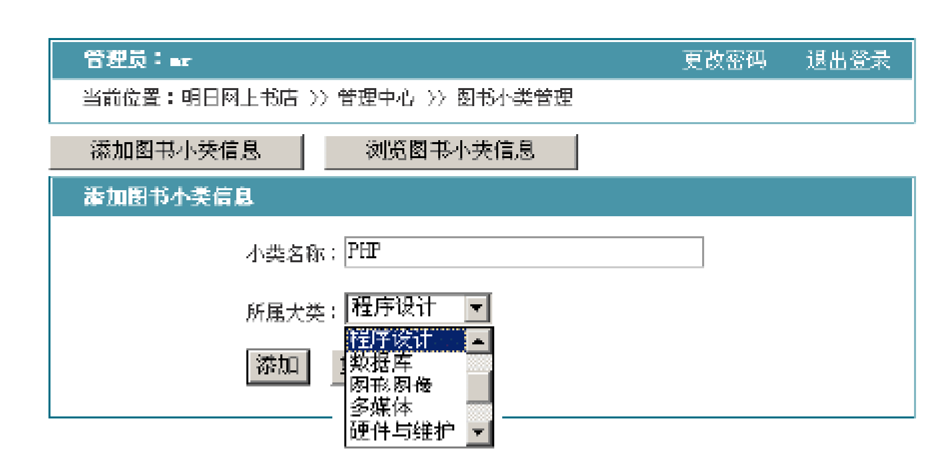


图 15.49　图书小类录入表单

关键技术

在图书小类管理中，应用最多的还是数据库管理类中的 executeSQL()方法，由其完成图书小类数据的添加、更新、查询和删除操作。同时还应用 Smarty 模板的 html_options 标签将所有大类名称在下拉列表中输出，其代码如下：

```
<select name="bigtypeid">
<option value="">-请选择-</option>
{html_options options = $arrayBtypeOption}
</select>
```

设计过程

（1）用户首先在小类名称文本框中输入小类名称，然后选择该小类所属大类，单击“添加”按钮，表单内容将被提交到 admin-smalltype.php 文件中进行保存，admin-smalltype.php 的代码如下：

```
$btypes = $adminDB->executeSQL("select id, typename from tb_bigtype order by addtime", $connID);    //查询大类信息
for ($i = 0; $i < count($btypes); $i ++) {                                                          //将大类信息保存到数组中
    $arrayBtypeOption[$btypes[$i]['id']] = $btypes[$i]['typename'];
}
$smarty->assign('arrayBtypeOption', $arrayBtypeOption);
if (isset($_POST['typename']) && $_POST['typename'] != '') {
    if (! $adminDB->executeSQL("select id, typename from tb_smalltype where typename='" . trim($_POST['typename']) . "'", $connID)) {
        if (! $adminDB->executeSQL("insert into tb_smalltype(typename, bigtypeid, addtime) values('" . trim($_POST['typename']) . "', '" .
trim($_POST['bigtypeid']) . "','" . date('Y-m-d H:i:s') . "')", $connID)) {                        //执行添加操作
            echo "<script>alert('类别添加失败！');</script>";
        } else {
            echo "<script>alert('类别添加成功！');</script>";
        }
    } else {
        echo "<script>alert('该类别已经添加！');</script>";
    }
}
```

上述代码首先查询所有图书大类名称和 ID 并保存到二维数组中，然后通过 Smarty 模板的 html_options 标签将所有大类名称在下拉列表框中输出，然后使用函数 isset()判断是否提交表单，如果是，则使用数据库管理类的 executeSQL()方法保存小类信息。

（2）在图书小类展示模块中，主要显示出图书小类的名称、所属大类和小类添加时间，其运行效果如图 15.50 所示。

浏览图书小类信息

类别名称	所属大类	添加时间	操作
C++/VC	程序设计	2009-05-21 10:23:37	
ASP	程序设计	2009-05-21 10:23:41	
PHP	程序设计	2009-05-21 10:26:14	
JAVA/JSP	程序设计	2009-05-21 10:26:18	

图 15.50　图书小类信息列表

单击图书小类信息列表页中的编辑按钮，将出现小类编辑表单，用户更改小类信息并提交表单后，表单内容将被提交到 admin-listsmalltype.php 文件中实现对图书小类信息的更改，其关键代码如下：

```
$isShow = 'F';
if ((isset($_GET['f']) && $_GET['f'] == 'edit') || (isset($_POST['f']) && $_POST['f'] == 'edit')) {
    $btypes = $adminDB->executeSQL("select id, typename from tb_bigtype order by addtime", $connID);    //获得大类信息
    for ($i = 0; $i < count($btypes); $i ++) {                                                          //将大类信息保存到数组中
        $arrayBtypeOption[$btypes[$i]['id']] = $btypes[$i]['typename'];
    }
    $smarty->assign('arrayBtypeOption', $arrayBtypeOption);
    if (isset($_POST['f']) && $_POST['f'] == 'edit') {                                                  //进行编辑操作
        if (! $adminDB->executeSQL("update tb_smalltype set typename='" . $_POST['typename'] . "', bigtypeid='" . $_POST['bigtypeid'] . "'    where
id='" . $_POST['id'] . "'", $connID)) {
            echo "<script>alert('类别更改失败！');</script>";
        } else {
            echo "<script>alert('类别更改成功！');</script>";
        }
    }
    if (isset($_GET['id'])) {
        $id = $  GET['id'];
    } else {
        $id = $_POST['id'];
    }
    $isShow = 'T';
    $smalltype = $adminDB->executeSQL("select id, typename, bigtypeid from tb_smalltype where id='" . $id . "'", $connID);    //查询小类信息
    $smarty->assign('smalltype', $smalltype);
}
```

上述代码首先判断$_GET['f']数组元素是否已经设置，如果是，则说明用户单击编辑按钮，这时查询出所有大类名称，并使当前所编辑小类的大类名称处于选中状态，然后判断是否设置$_POST['f']数组元素的值，如果是，则说明已经提交小类信息更改表单，此时通过数据库管理类的 executeSQL()方法更改小类信息。

（3）如果管理员单击小类信息列表页中的删除按钮，则首先弹出一个提示框提示用户是否确认删除，如果是，则通过数据库管理类的 executeSQL()方法执行 delete 语句删除指定的小类信息，其代码如下：

```
if (isset($_GET['f']) && $_GET['f'] == 'del') {
    if (! $adminDB->executeSQL("delete from tb_smalltype where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('类别删除失败！');</script>";                                              //执行删除
    }
}
```

秘笈心法

心法领悟 618：应用 isset()函数判断指定的变量是否存在。

isset()函数检查变量是否被设置，如果设置则返回 TRUE，否则返回 FALSE。由于这是一个语言结构而非函数，因此它无法被“变量函数”调用，其语法如下：

```
bool isset ( mixed var [, mixed var [, ...]])
```

参数 var 为被检查的变量，可以有多个。

实例 619　**出版社分类管理**

光盘位置：光盘\MR\15\619

高级

趣味指数：★★★☆

实例说明

图书不但可以按内容划分，还可以按其所属出版社进行分类，这样可以更加方便地引导读者选购自己信赖



的出版社所出版的图书。为使出版社分类更加形象、易于识别，在出版社分类中增加了出版社的 Logo 图片信息，浏览出版社类别信息的页面效果如图 15.51 所示。



图 15.51　浏览出版社类别信息

关键技术

在出版社分类管理中，关键是出版社 Logo 图标上传的实现方法，其上传应用的是 move_uploaded_file()函数。

（1）在执行上传之前，应用 is_dir()函数判断指定的文件夹是否存在，如果不存在则应用 mkdir()函数创建一个新文件夹。

（2）应用$_FILES[]全局数组获取上传图片的原始名称。

（3）定义上传文件在服务器文件夹下的存储名称，应用 date()函数获取时间戳，应用 mt_rand()函数定义随机数，通过 substr()函数获取原始文件的后缀名称，从而组成新的文件名称。

（4）定义上传文件在服务器中的存储路径。

（5）应用 move_uploaded_file()函数执行文件上传操作。

上述函数的详细讲解可以参考本书第 4 章的内容，这里不再赘述。

设计过程

（1）网站管理者在添加出版社名称并选择出版社 Logo 图片后，单击“添加”按钮即可将出版社类别信息提交到 admin-pub.php 文件中进行处理，其代码如下：

```
if (isset($_POST['pubname']) && $_POST['pubname'] != '') {                    //如果提交了表单
                                                                              //判断该出版社是否已经添加
    if (! $adminDB->executeSQL("select id, pubname from tb_pub where pubname='" . trim($_POST['pubname']) . "'", $connID)) {
        if (isset($_FILES["pubimg"]["name"]) && $_FILES["pubimg"]["name"] != "") {   //上传出版社 Logo
            $dir = "./upfiles/bookimg";
            if (! is_dir($dir)) {
                mkdir($dir);
            }
            $upfilename = $_FILES["pubimg"]["name"];
            $filename = date("YmdHis") . mt_rand(1000, 9999) . substr($upfilename, strpos($upfilename, "."), strlen($upfilename) -
strpos($upfilename, "."));                                                    //设置出版社保存文件名
            $address = $dir . "/" . $filename;
            @move_uploaded_file($_FILES["pubimg"]["tmp_name"], $address);     //执行上传操作
        } else {
            $filename = "";
        }
        if (! $adminDB->executeSQL("insert into tb_pub(pubname, pubimg, addtime) values('" . trim($_POST['pubname']) . "', '" . $filename . "', '" .
date('Y-m-d H:i:s') . "')", $connID)) {                                       //保存出版社类别信息
            echo "<script>alert('出版社类别添加失败！');</script>";
        } else {
            echo "<script>alert('出版社类别添加成功！');</script>";
        }
    } else {
        echo "<script>alert('该出版社类别已经添加！');</script>";
    }
}
```

实例 620　图书信息管理

光盘位置：光盘\MR\15\620

高级

趣味指数：★★★★☆

实例说明

明日科技图书网的定位为图书产品的展示和销售，所以图书信息管理模块非常关键，该项目的图书信息管理模块主要包括对图书信息的添加、更改、删除和查询等操作。在图书信息添加页面中，添加的内容包括图书的名称、类别、页数、字数、作者、ISBN 号和价格等，其运行效果如图 15.52 所示。



图 15.52　图书信息添加

关键技术

在图书信息管理中，非常关键的一点是明确数据表的各个字段代表的含义。在明日科技图书网的前台中，在对图书信息进行分类时，依据的就是图书信息表中某个字段的值，例如，判断图书是否为新书、是否特价等。所以必须对图书信息表中各个字段了如指掌才能更好地完成判断操作。tb_bookinfo 图书信息表的说明如表 15.2 所示。

表 15.2　tb_bookinfo 图书信息表

字段名称	数据类型	字段大小	是否主键	说　明
id	int	11	主键	自动增长
bookname	varchar	50		大类名称
smalltypeid	datetime			添加时间
oldprice	float			图书原价
newprice	float			本站价格
bookimg	varchar	100		图书封面图片名称
isnew	bool			是否新书
addtime	datetime			添加时间
about	text			图书介绍
issepprice	bool			是否特价图书
browsertime	int	11		浏览次数

续表

字段名称	数据类型	字段大小	是否主键	说明
ismrbooktj	bool			是否**科技创作
pubid	int	11		出版社 ID
writer	varchar	50		作者
pubtime	date			出版时间
ishave	bool			是否有货
bookcc	int	2		图书层次标识
page	int	4		图书页码
isbn	varchar	20		图书 ISBN 号
zs	int	4		图书字数
bc	varchar	50		图书版权
directory	text			图书目录
bookids	varchar	100		关联图书 ID 所组成的字符串
ishotsell	bool			是否热卖图书
isterm	bool			是否期书

设计过程

（1）管理员首先在图 15.52 所示的页面中录入图书信息，然后单击“添加”按钮，则图书信息将被提交到 admin-book.php 文件中并进行保存，其代码如下：

```
if (isset($_POST['bookname']) && $_POST['bookname'] != '' && $_POST['change'] == 'F') {
    if (! $adminDB->executeSQL("select id, bookname from tb_bookinfo where bookname='" . trim($_POST['bookname']) . "'", $connID)) {
        //判断图书信息是否已经被添加
        if (isset($_FILES["bookimg"]["name"]) && $_FILES["bookimg"]["name"] != "") {
            $dir = "./upfiles/bookimg";
            if (! is_dir($dir)) {
                mkdir($dir);
            }
            $upfilename = $_FILES["bookimg"]["name"];
            $filename = date("YmdHis") . mt_rand(1000, 9999) . substr($upfilename, strpos($upfilename, "."), strlen($upfilename) -
strpos($upfilename, "."));                                                    //设置图书封面名称
            $address = $dir . "/" . $filename;
            @move_uploaded_file($_FILES["bookimg"]["tmp_name"], $address);                 //上传图书封面
        } else {
            $filename = "";
        }
        $pubtime = $_POST['pyear'] . '-' . $_POST['pmonth'] . '-00';
        if (! $adminDB->executeSQL("insert into tb_bookinfo(bookname, smalltypeid, pubid, page, zs, isbn, bc, writer, pubtime, oldprice, newprice,
bookcc, bookids, bookimg, directory, about, isnew, issepprice, ishotsell, isterm ,ismrbooktj ,ishave, addtime, browsertime) values('" .
trim($_POST['bookname']) . "', '" . trim($_POST['smalltypeid']) . "', '" . trim($_POST['pubid']) . "', '" . trim($_POST['page']) . "', '" . trim($_POST['zs']) .
"', '" . trim($_POST['isbn']) . "', '" . trim($_POST['bc']) . "', '" . trim($_POST['writer']) . "', '" . $pubtime . "', '" . trim($_POST['oldprice']) . "', '" .
trim($_POST['newprice']) . "', '" . trim($_POST['bookcc']) . "', '" . trim($_POST['bookids']) . "', '" . $filename . "', '" . trim($_POST['directory']) . "', '" .
trim($_POST['about']) . "', '" . trim($_POST['isnew']) . "', '" . trim($_POST['issepprice']) . "', '" . trim($_POST['ishotsell']) . "', '" . trim($_POST['isterm']) .
"', '" . trim($_POST['ismrbooktj']) . "', '" . trim($_POST['ishave']) . "', '" . date('Y-m-d H:i:s') . "', '0')", $connID)) {      //保存图书信息
            echo "<script>alert('图书添加失败！');</script>";
        } else {
            echo "<script>alert('图书添加成功！');</script>";
        }
    } else {
        echo "<script>alert('该图书已经添加！');</script>";
    }
}
$isEdit = 'F';
if (isset($_GET['f']) && $_GET['f'] == 'edit' || isset($_POST['change']) && $_POST['change'] == 'T') {
    if (isset($_POST['change']) && $_POST['change'] == 'T') {
        if (isset($_FILES["bookimg"]["name"]) && $_FILES["bookimg"]["name"] != "") {
            $dir = "./upfiles/bookimg";
```

```
            if (! is_dir($dir)) {
                mkdir($dir);
            }
            $upfilename = $_FILES["bookimg"]["name"];
            $filename = date("YmdHis") . mt_rand(1000, 9999) . substr($upfilename, strpos($upfilename, "."), strlen($upfilename) -
strpos($upfilename, "."));
            ;
            $address = $dir . "/" . $filename;
            @move_uploaded_file($_FILES["bookimg"]["tmp_name"], $address);
        } else {
            $tmpbook = $adminDB->executeSQL("select bookimg from tb_bookinfo where id='" . $_POST['id'] . "'", $connID);
            $filename = $tmpbook[0]['bookimg'];
        }
        $pubtime = $_POST['pyear'] . '-' . $_POST['pmonth'] . '-00';
        ...//省略用户设置提交内容的代码，请参见本书光盘中源码
        if (! $adminDB->executeSQL("update tb_bookinfo set bookname='" . trim($_POST['bookname']) . "', smalltypeid='" .
trim($_POST['smalltypeid']) . "', pubid='" . trim($_POST['pubid']) . "', page='" . trim($_POST['page']) . "', zs='" . trim($_POST['zs']) . "', isbn='" .
trim($_POST['isbn']) . "', bc='" . trim($_POST['bc']) . "', writer='" . trim($_POST['writer']) . "', pubtime='" . $pubtime . "', oldprice='" .
trim($_POST['oldprice']) . "', newprice='" . trim($_POST['newprice']) . "', bookcc='" . trim($_POST['bookcc']) . "', bookids='" . trim($_POST['bookids']) .
"', bookimg='" . $filename . "', directory='" . trim($_POST['directory']) . "', about='" . trim($_POST['about']) . "', isnew='" . $isnew . "', issepprice='" .
$issepprice . "', ishotsell='" . $ishotsell . "', isterm='" . $isterm . "' ,ismrbooktj='" . $ismrbooktj . "' ,ishave='" . $ishave . "' where id='" . $_POST['id'] . "'",
$connID)) {                                                                                    //更改图书信息
            echo "<script>alert('图书信息更改失败 ！ ');</script>";
        } else {
            echo "<script>alert('图书信息更改成功 ！ ');</script>";
        }
    }
    if (isset($_POST['id'])) {
        $id = $_POST['id'];
    } else {
        $id = $_GET['id'];
    }
    $isEdit = 'T';
    $book = $adminDB->executeSQL("select id, bookname, smalltypeid, pubid, page, zs, isbn, bc, writer, pubtime, oldprice, newprice, bookcc, bookids,
bookimg, directory, about, isnew, issepprice, ishotsell, isterm ,ismrbooktj ,ishave, addtime, browsertime from tb_bookinfo where id='" . $id . "' ",
$connID);                                                                                      //查询图书信息
    $smarty->assign('book', $book);
    $smarty->assign('pyear', substr($book[0]['pubtime'], 0, 4));
    $smarty->assign('pmonth', substr($book[0]['pubtime'], 5, 2));
}
```

在上述代码中，定义图书信息添加和更新两种操作。首先判断$_POST['change']数组元素的值，如果该值为F，则进行添加操作，通过数据库管理类的 executeSQL()方法执行 insert 语句将图书信息保存到数据库中，同时使用函数 move_uploaded_file()将图书封面上传到服务器中；如果$_POST['change']的值为 T，则执行图书信息的更改操作，在更改图书信息时，首先判断是否重新选择图书封面，如果是，则重新上传图书封面，然后执行 update 语句对图书信息进行更新。

（2）图书信息列表主要对图书名称、出版社、作者和价格等进行分页显示，其运行效果如图 15.53 所示。

浏览图书信息

图书名称	出版社	作者	市场价（元）	会员价（元）	操作
《PHP开发典型模块大全》	人民邮电出版社	邹天思 潘凯华 孙鹏	79.00	79.00	
《Visual C++开发经验技巧宝典》	人民邮电出版社	明日科技	89.00	89.00	
《Visual C++程序设计自学手册》	人民邮电出版社	明日科技 宋坤 刘锐宁	52.00	52.00	
《SQL Server 2005开发技术大全》	人民邮电出版社	明日科技	79.00	79.00	
《C#程序设计标准教程》	人民邮电出版社	明日科技 王小科 吕双	59.00	59.00	
《ASP.NET网络编程标准教程》	人民邮电出版社	张跃廷 丁红兵	55.00	55.00	

图 15.53　图书信息列表

通过图书信息列表页可以实现对图书信息的更改和删除操作，在讲解图书添加操作中已经介绍了图书信息更改的实现方法，这里不再赘述。当单击图书信息列表中的删除按钮时，首先弹出提示对话框提示用户是否确认删除，如果是，则通过如下代码执行对图书信息的删除操作：

```
if (isset($_GET['f']) && $_GET['f'] == 'del') {
    if (! $adminDB->executeSQL("delete from tb_bookinfo where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('图书信息删除失败！ ');</script>";                          //执行删除操作
    }
}
```

（3）与其他管理模块相比，图书信息量较大，为方便对图书信息进行管理，在图书信息管理模块中增加了图书信息查询功能，其运行效果如图 15.54 所示。

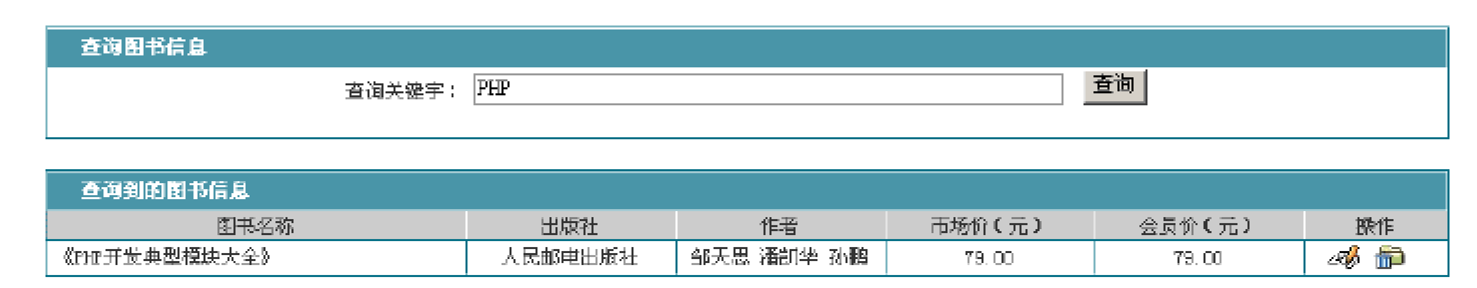


图 15.54　图书信息查询

首先在文本框中输入要查询的关键字，然后单击“查询”按钮，这时查询表单的关键字将被提交到 admin-searchbook.php 文件中，在该文件完成对图书信息的查询操作，并且将查询结果赋给模板变量，在模板页 admin-searchbook.phtml 中输出查询结果，其代码如下：

```
$isFind = 'F';
if (isset($_POST['bookname']) && $_POST['bookname'] != '' || isset($_GET['bookname']) && $_GET['bookname'] != '') {    //判断是否提交了表单
    if (isset($_POST['bookname'])) {                                                                                     //获取要查询的图书名称
        $bookname = $_POST['bookname'];
    } else {
        $bookname = $_GET['bookname'];
    }
    $books = $adminDB->executeSQL("select tb_bookinfo.id , tb_bookinfo.oldprice, tb_bookinfo.newprice, tb_bookinfo.bookname,
tb_bookinfo.writer,   tb_pub.pubname from tb_bookinfo, tb_pub where tb_bookinfo.pubid = tb_pub.id and bookname like '%" . $bookname . "%'   order
by tb_bookinfo.addtime desc", $connID);                                                                                  //执行模糊查询
    $smarty->assign('books', $books);
    $isFind = 'T';
    $smarty->assign('bookname', $bookname);
}
```

上述代码首先使用函数 isset()判断$_POST['bookname']数组元素是否已经被设置，如果是，则使用数据库管理类的 executeSQL()方法执行查询操作。

秘笈心法

心法领悟 620：通过 like 关键字执行模糊查询，查询图书的名称。

在本实例的图书查询中，应用 like 关键字执行多表模糊查询，查询图书的名称，并且将查询结果返回到模板变量中。

实例 621	图书试读管理 光盘位置：光盘\MR\15\621	高级 趣味指数：★★★☆

实例说明

为了让读者选择一本真正适合自己的图书并且吸引更多读者购买，在制作明日科技图书网时增加了图书试读模块，这样读者在购买图书前就可以大概了解所购买图书的内容，非常人性化。在图书试读信息添加页面中包括图书类别下拉列表框、图书名称选择下拉列表框和试读文件名称输入文本框，其运行效果如图 15.55 所示。

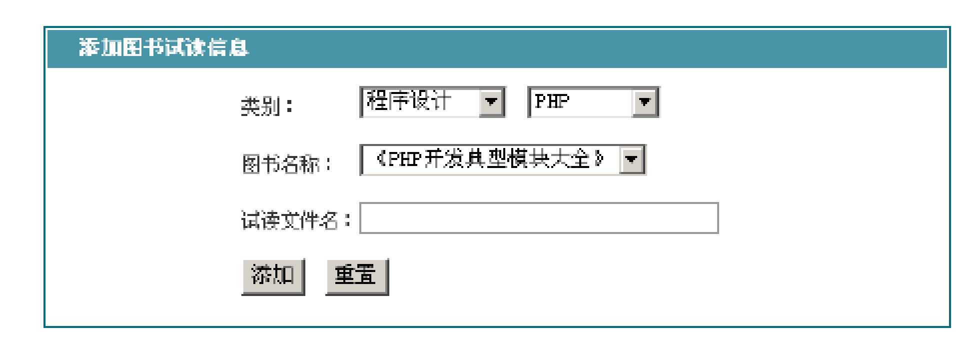
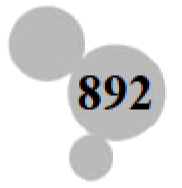


图 15.55　添加图书试读信息页面

关键技术

在添加图书试读信息的模板页 admin-read.phtml 中，通过 Smarty 模板中的{ldelim} {rdelim}标签直接在模板页中编写 JavaScript 脚本，应用 jQuery 技术实现下拉菜单的三级联动，其关键代码如下：

```
<script src="{util->baseUrl}/js/jquery.js"></script>
```



```
<script language="javascript" >
$(document).ready(
    function(){ldelim}
      $("#bigtypeid").change(function(){ldelim}
            $.get("{util->baseUrl}/getstype.php?stid="+$("#bigtypeid").val(), null, function(data){ldelim}
                  $("#smalltypeid").empty();
                  $("#smalltypeid").append(data);
          {rdelim});
        {rdelim});

          $("#smalltypeid").change(function(){ldelim}
            $.get("{util->baseUrl}/getbook.php?bid="+$("#smalltypeid").val(), null, function(data){ldelim}
                  $("#bookid").empty();
                  $("#bookid").append(data);
          {rdelim});
        {rdelim});
    {rdelim}
);
</script>
```

设计过程

（1）管理员在模板页 admin-read.phtml 中输入图书 3 试读信息，将 form 表单中的数据提交到 admin-read.php 文件中进行保存。form 表单的内容如下：

```
<form name="form_read" method="post" action="{util->baseUrl}/admin-read.php" onsubmit="return chkInputRead(this)">
<div style="width:100%; text-align:left; padding-top:5px; padding-left:50px">类别：      
<select id="bigtypeid" name="bigtypeid">
                <option value='' selected>-请选择-</option>
                {html_options options = $arrayBtypeOption}
    </select> 
    <select name="smalltypeid" id="smalltypeid">
                <option value='' selected>-请选择-</option>
    </select> <br /><br />
    图书名称：  
    <select name="bookid" id="bookid">
                <option value='' selected>-请选择-</option>
    </select> <br /><br />
    试读文件名：<input type="text" name="filename" size="30" class="input"/><br /><br />
                      <input type="submit" value="添加" />  <input type="reset" value="重置" />
</div>
</form>
```

（2）新建 admin-read.php 文件，保存表单中提交的试读信息数据，其代码如下：

```
$btypes = $adminDB->executeSQL("select id, typename from tb_bigtype order by addtime", $connID);          //查询图书大类
for ($i = 0; $i < count($btypes); $i ++) {                                                             //将大类保存到数组中
    $arrayBtypeOption[$btypes[$i]['id']] = $btypes[$i]['typename'];
}
$smarty->assign('arrayBtypeOption', $arrayBtypeOption);
if (isset($_POST['filename']) && $_POST['filename'] != '') {                                           //判断是否提交了表单
    if (! $adminDB->executeSQL("select id, filename from tb_read where bookinfoid='" . $_POST['bookid'] . "'", $connID)) {
        //判断该书的试读信息是否已经添加
        if (! $adminDB->executeSQL("insert into tb_read(bookinfoid, filename,   addtime) values('" . $_POST['bookid'] . "', '" . $_POST['filename'] .
"', '" . date('Y-m-d H:i:s') . "')", $connID)) {
            echo "<scriipt>alert('试读添加失败！');</script>";
        } else {
            echo "<script>alert('试读添加成功！');</script>";
        }
    } else {
        echo "<script>alert('已经为该图书添加试读！');</script>";
    }
}
```

（3）试读信息列表页主要以分页的形式显示出试读图书的名称、试读文件名和添加时间等信息，其运行效果如图 15.56 所示。

浏览图书试读信息

图书名称	试读文件名	添加时间	操作
《Visual C++开发经验技巧宝典》	1.doc	2009-07-03 17:45:13	
《Visual C++程序设计自学手册》	1.doc	2009-07-03 17:44:38	

共 2 本　每页显示 20 本　第 1 页/共 1 页　　首页　上一页　下一页　尾页

图 15.56　图书试读信息列表

单击试读信息列表的编辑按钮将出现试读信息更改表单，在该表单中修改试读图书的内容，将表单数据提交到 admin-listread.php 页面中进行修改，其代码如下：

```
if ((isset($_GET['f']) && $_GET['f'] == 'edit') || (isset($_POST['f']) && $_POST['f'] == 'edit')) {          //判断是否提交了表单
    if (isset($_POST['f']) && $_POST['f'] == 'edit') {
        if (! $adminDB->executeSQL("update tb_read set bookinfoid='" . $_POST['bookid'] . "', filename='" . $_POST['filename'] . "'   where id='" .
$_POST['id'] . "'", $connID)) {     //执行更改操作
            echo "<script>alert('试读更改失败！');</script>";
        } else {
            echo "<script>alert('试读更改成功！');</script>";
        }
    }
    if (isset($_GET['id'])) {
        $id = $_GET['id'];
    } else {
        $id = $_POST['id'];
    }
    $isShow = 'T';
    $read = $adminDB->executeSQL("select tb_read.id, tb_read.filename, tb_bookinfo.bookname, tb_read.addtime from tb_read, tb_bookinfo where
tb_read.bookinfoid = tb_bookinfo.id and tb_read.id='" . $id . "'", $connID);  //查询试读信息
    $smarty->assign('read', $read);
}
```

单击试读信息的删除按钮，会弹出是否确认删除的对话框，如果选择确认删除，则执行如下代码实现试读信息的删除：

```
if (isset($_GET['f']) && $_GET['f'] == 'del') {
    if (! $adminDB->executeSQL("delete from tb_read where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('试读删除失败！');</script>";   //执行删除试读操作
    }
}
```

秘笈心法

心法领悟 621：分页功能的实现。

在本实例中，试读信息的分页输出应用的是分页类中的 pageData()方法，将分页查询结果赋给模板变量，在模板页中创建分页超链接，完成试读信息的分页输出。

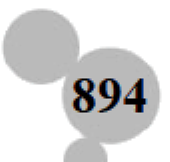
15.12　用 户 管 理

用户管理模块实现的功能包括对网站注册用户的管理和对用户反馈信息的管理，本节将讲解这两个功能的实现过程。

实例 622	用户信息管理 光盘位置：光盘\MR\15\622	高级 趣味指数：★★★☆

实例说明

在用户信息管理模块中对网站注册用户进行分页输出，输出内容包括用户昵称、真实姓名、性别和联系电话等，其运行效果如图 15.57 所示。



实例 623 用户反馈管理

光盘位置：光盘\MR\15\623

高级

趣味指数：★★★☆

实例说明

用户反馈管理模块输出用户对网站本身、网站中的某本图书，或者其他方面的一些建议、意见等，其输出效果如图 15.58 所示。

浏览用户反馈信息

主题	用户	发表时间	操作
程序开发经验交流	mr	2010-08-27 02:35:37	
我的建议	mr	2010-05-13 07:52:42	
用户意见	2000	2009-08-13 12:28:10	

共 3 条　每页显示 20 条　第 1 页/共 1 页　首页　上一页　下一页　尾页

查看用户反馈详细信息

用户昵称：mr　发表时间：2010-05-13 07:52:42

发表时间：我的建议

我的建议

图 15.58　浏览用户反馈信息

关键技术

本实例实现的功能与实例 622 是相同的，唯一的不同之处是本实例操作的是 tb_feedback 数据表，而用户信息管理实例操作的是 tb_user 表。

设计过程

（1）浏览用户反馈信息由 admin-listuserfeedback.php 和 admin-listuserfeedback.phtml 两个文件组成。在 admin-listuserfeedback.php 文件中，首先判断指定的分页变量是否存在，然后根据$_GET[]获取的变量值进行判断，执行用户反馈信息的删除操作、数据的分页输出或者查看指定用户反馈的详细信息，最后指定模板页，其代码如下：

```
<?php
require_once 'admin-header.php';                                    //包含头文件
if (! isset($_GET['page']) || $_GET['page'] == '') {                //判断分页变量是否存在
      $page = '1';
} else {
      $page = $_GET['page'];
}
if (isset($_GET['f']) && $_GET['f'] == 'del') {                     //判断是否执行删除操作
      if (! $adminDB->executeSQL("delete from tb_feedback where id='" . $_GET['id'] . "'", $connID)) {
            echo "<script>alert('用户反馈信息删除失败！');</script>";
      }
}
$sql = "select id, usernc, title, addtime from tb_feedback order by addtime desc";
$feedbacks = $pageDB->pageData($sql, $connID, 20, $page);           //执行分页查询
$smarty->assign('feedbacks', $feedbacks);
$isShow = 'F';
if (isset($_GET['f']) && $_GET['f'] == 'edit') {                    //判断是否执行详细信息查询操作
      $userfeed = $adminDB->executeSQL("select id, usernc, title, content, addtime from tb_feedback where id='" . $_GET['id'] . "'", $connID);
      $isShow = 'T';
      $smarty->assign('userfeed', $userfeed);                       //将查询结果赋给模板变量
}
$smarty->assign('isShow', $isShow);
$smarty->display('admin-listuserfeedback.phtml');                   //指定模板页
require_once 'admin-footer.php';                                    //包含尾文件
```

（2）新建 admin-listuserfeedback.phtml 模板页。首先通过 section 语句循环输出用户反馈信息，并且创建查看指定用户反馈的详细信息和删除指定用户的超链接。然后定义分页超链接，完成数据的分页输出。最后根据



续表

字 段 名 称	数 据 类 型	字 段 大 小	是 否 主 键	说　明
numstr	varchar	100		订购图书数量组成字符串
goodsprice	float			图书价格
yjprice	float			邮寄价格
totalprice	float			总价格
rectype	varchar	20		收货方式
paytype	varchar	20		支付方式
addtime	datetime			订购时间
isfk	bool			是否付款
isfh	bool			是否发货
issh	bool			是否收货
isqx	bool			是否被用户取消订购

设计过程

用户订单管理首先完成用户订单信息的分页输出，然后执行订单支付状态的更改、订单信息的查看和删除等操作。

（1）在 admin-listorder.php 文件中，首先包含头文件，根据传递的查询字符串 page 的值并通过数据库分页类的 pageData()方法实现订单信息的分页输出。然后通过数据库管理类的 executeSQL()方法，执行 update 语句实现订单状态的更改。最后通过数据库管理类的 executeSQL()方法执行 select 语句，查询指定订单的详细信息，并且将查询结果在模板页中输出。admin-listorder.php 文件的代码如下：

```
<?php
require_once 'admin-header.php';                                    //包含头文件
if (! isset($_GET['page']) || $_GET['page'] == '') {                 //判断分页变量的值
    $page = '1';
} else {
    $page = $_GET['page'];
}
if (isset($_GET['f']) && $_GET['f'] == 'del') {                      //判断是否执行删除操作
    if (! $adminDB->executeSQL("delete from tb_order where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('订单信息删除失败！ ');</script>";
    }
}
if (isset($_GET['c']) && $_GET['c'] != '') {                         //更改订单状态
    $c = $_GET['c'];
    if ($c == 'isfk') {                                              //判断是否已经付款
        $adminDB->executeSQL("update tb_order set isfk = !isfk where id='" . $_GET['id'] . "'", $connID);
    } elseif ($c == 'isfh') {                                        //判断是否已经发货
        $adminDB->executeSQL("update tb_order set isfh = !isfh where id='" . $_GET['id'] . "'", $connID);
    } elseif ($c == 'issh') {                                        //判断是否已经收货
        $adminDB->executeSQL("update tb_order set issh = !issh where id='" . $_GET['id'] . "'", $connID);
    }
}
$sql = "select id, orderno, username, tel, goodsprice, yjprice, totalprice, isfk, isfh, issh, isqx from tb_order order by addtime desc";
//定义 SQL 语句
$orders = $pageDB->pageData($sql, $connID, 20, $page);               //执行分页查询
$smarty->assign('orders', $orders);                                  //将查询结果赋给模板变量
$isShow = 'F';
if (isset($_GET['f']) && $_GET['f'] == 'edit') {                     //判断是否执行查看订单详细信息的操作
    $order = $adminDB->executeSQL("select id, orderno, username, address, sex, yb, tel, idstr, numstr, rectype, paytype, goodsprice, yjprice, totalprice,
addtime, isfk, isfh, issh from tb_order where id='" . $_GET['id'] . "'", $connID);       //执行查询
    $arrayIds = explode('@', $order[0]['idstr']);
    $arrayNums = explode('@', $order[0]['numstr']);
    $arrayCarInfos = array();
    $totalPrice = 0;
    for ($i = 0; $i < count($arrayIds); $i ++) {
```

```
            $bookid = $arrayIds[$i];
            if ($bookid != '') {
                $tmpArray = array();
                $bookinfo = $adminDB->executeSQL("select id, bookname ,oldprice, newprice from tb_bookinfo where id='" . $bookid . "'", $connID);
                $tmpArray['id'] = $bookinfo[0]['id'];
                $tmpArray['bookname'] = $bookinfo[0]['bookname'];
                $tmpArray['oldprice'] = $bookinfo[0]['oldprice'];
                $tmpArray['newprice'] = $bookinfo[0]['newprice'];
                $tmpArray['num'] = $arrayNums[$i];
                $tmpArray['smallTotalPrice'] = $bookinfo[0]['newprice'] * $arrayNums[$i];
                $totalPrice += $tmpArray['smallTotalPrice'];
                array_push($arrayCarInfos, $tmpArray);
            }
        }
        $smarty->assign('arrayCarInfos', $arrayCarInfos);                //将生成的订单详细信息赋给模板变量
        $smarty->assign('order', $order);
        $isShow = 'T';
}
$smarty->assign('isShow', $isShow);                                      //通过该变量值判断是否输出订单详细信息
$smarty->display('admin-listorder.phtml');                              //指定模板页
require_once 'admin-footer.php';                                         //包含尾文件
```

（2）在 admin-listorder.phtml 模板页中，首先通过 section 语句循环输出订单信息，创建复选框更改用户订单的状态，创建查看订单详细信息和删除指定订单的超链接。然后创建数据分页超链接，完成订单信息的分页输出。最后输出模板变量中存储的指定订单的详细信息。其关键代码如下：

```
<div style="border-right:1px solid #006D84;{if $orders.data[oID].isqx==1}padding-top:4px{/if}">
{if $orders.data[oID].isqx==1}--已取消--{else}付款：<input type="checkbox" name="isfk" {if $orders.data[oID].isfk==1}checked{/if}
onclick="window.location.href='admin-listorder.php?c=isfk&id={$orders.data[oID].id}'" />  
发货：<input type="checkbox" name="isfh" {if $orders.data[oID].isfh==1}checked{/if}
onclick="window.location.href='admin-listorder.php?c=isfh&id= {$orders.data[oID].id}'">  
收货：<input type="checkbox" name="issh" {if $orders.data[oID].issh==1}checked{/if}
onclick="window.location.href='admin-listorder.php?c=issh&id= {$orders.data[oID].id}'">
{/if}
</div>
<div style="width:9%; height:20px; float:left; padding-top:2px">
<a href="{util->baseUrl}/admin-listorder.php?f=edit&id={$orders.data[oID].id}"><img src="{util->baseUrl}/img/edit.gif" border="0"/></a> 
<a href="javascript:if(window.confirm('确定删除？')==true){ldelim}window.location.href='{util->baseUrl}/admin-listorder.php?f=del&id=
{$orders.data[oID].id}';{rdelim}"><img src="{util->baseUrl}/img/del.gif" border="0"/></a>
</div>
```

秘笈心法

心法领悟 624：通过复选框完成订单状态的更改。

在 admin-listorder.phtml 模板页中创建复选框，通过 onclick 事件调用 JavaScript 脚本，在 JavaScript 脚本中，应用 window 对象的 location 方法调用 admin-listorder.php 文件，并传递两个参数 c 和 id，在 admin-listorder.php 文件中完成订单状态的更改操作。

15.14　通用信息管理

在通用信息管理模块可实现对新闻公告、网上畅销书讯和明日图书排行管理，本节将讲解其功能的具体实现方法。

实例 625	新闻公告管理 光盘位置：光盘\MR\15\625	高级 趣味指数：★★★★☆

实例说明

在新闻公告管理模块中，可实现新闻公告的添加、浏览、更新和删除操作。其中在添加新闻公告功能中，



```
	if(isset($_POST['id']) && $_POST['id']!=''){
		if(!$adminDB->executeSQL("update tb_tell set title='".$_POST['title']."', content='".$_POST['content']."' where id='".$_POST['id']."'",
$connID)){
			echo "<script>alert('更改失败！');</script>";
		}else {
			echo "<script>alert('更改成功！');</script>";
		}
	}
$isEdit = 'T';
if(isset($_POST['id'])){                                                          //判断提交的 ID 值是否为空
	$id = $_POST['id'];
}else{
	$id = $_GET['id'];
}
$info = $adminDB->executeSQL("select id, title, content from tb_tell where id='".$id."'", $connID);   //执行查询操作
$smarty->assign('info', $info);                                                   //将查询结果赋给模板变量
}
$smarty->assign('isEdit', $isEdit);
$smarty->display('admin-tell.phtml');                                             //指定模板页
require_once 'admin-footer.php';
```

（2）新闻公告浏览由 admin-listtell.php 和 admin-listtell.phtml 两个文件组成。在 admin-listtell.php 文件中，首先定义删除指定新闻公告的方法，然后执行新闻公告的分页查询，最后指定模板页，其关键代码如下：

```
<?php
require_once 'admin-header.php';                                                  //包含头文件
if(isset($_GET['f']) && $_GET['f'] == 'del'){
	if(!$adminDB->executeSQL("delete from tb_tell where id='".$_GET['id']."'", $connID)){
		echo "<script>alert('公告删除失败！');</script>";
	}
}
$sql = "select id, title, addtime from tb_tell order by addtime desc";           //定义分页查询的 SQL 语句
$tells = $pageDB->pageData($sql, $connID, 20, $page);                            //执行分页查询
$smarty->assign('tells', $tells);
$smarty->display('admin-listtell.phtml');                                         //指定模板页
require_once 'admin-footer.php';                                                  //包含尾文件
```

在 admin-listtell.phtml 模板文件中，通过 section 语句循环输出新闻公告信息，并且创建编辑和删除超链接，最后编写分页超链接，其关键代码如下：

```
<div style="width:10%; height:20px; float:left; padding-top:2px">
<a href="{util->baseUrl}/admin-tell.php?f=edit&id={$tells.data[tID].id}"><img src="{util->baseUrl}/img/edit.gif" border="0"/></a> 
<a href="javascript:if(window.confirm('确定删除？')==true){ldelim}window.location.href='{util->baseUrl}/admin-listtell.php?f=del&id=
{$tells.data[tID].id}';{rdelim}"><img src="{util->baseUrl}/img/del.gif" border="0"/></a>
</div>
```

秘笈心法

心法领悟 625：盘点本章实例。

本章实例均提取自明日科技网上书店这个完整项目，其目的是让读者了解前面章节中介绍的技术在实际项目开发中的应用，以此来巩固前面所学知识，真正做到理论与实践相结合，学以致用。